Preliminary

AM335x Sitara
ARM Microprocessors (MPUs)

Technical Reference Manual

Literature Number: SPRUH73
July 2011

DRAFT ONLY

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Contents

Preface **127**

1 Introduction **129**

2 Interrupts **131**

2.1 ARM Cortex A8 Interrupts 132
2.2 ARM Cortex M3 Interrupts 135
2.3 PRUSS Interrupts 137
2.4 EDMA Events 139
2.5 Crypto DMA Events 142
2.6 PWM Events 144
2.7 INTCPS Registers 144
2.7.1 INTCPS_REVISION Register (offset = 0h) [reset = 0h] 151
2.7.2 INTCPS_FIQ_PRIORITY Register (offset = 64h) [reset = 0h] 152
2.7.3 INTCPS_THRESHOLD Register (offset = 68h) [reset = 0h] 153
2.7.4 INTCPS_ITR0 Register (offset = 80h) [reset = 0h] 154
2.7.5 INTCPS_MIR0 Register (offset = 84h) [reset = 0h] 155
2.7.6 INTCPS_MIR_CLEAR0 Register (offset = 88h) [reset = 0h] 156
2.7.7 INTCPS_MIR_SET0 Register (offset = 8Ch) [reset = 0h] 157
2.7.8 INTCPS_ISR_SET0 Register (offset = 90h) [reset = 0h] 158
2.7.9 INTCPS_ISR_CLEAR0 Register (offset = 94h) [reset = 0h] 159
2.7.10 INTCPS_PENDING_IRQ0 Register (offset = 98h) [reset = 0h] 160
2.7.11 INTCPS_PENDING_FIQ0 Register (offset = 9Ch) [reset = 0h] 161
2.7.12 INTCPS_SYSCONFIG Register (offset = 10h) [reset = 0h] 162
2.7.13 INTCPS_ITR1 Register (offset = A0h) [reset = 0h] 163
2.7.14 INTCPS_MIR1 Register (offset = A4h) [reset = 0h] 164
2.7.15 INTCPS_MIR_CLEAR1 Register (offset = A8h) [reset = 0h] 165
2.7.16 INTCPS_MIR_SET1 Register (offset = ACh) [reset = 0h] 166
2.7.17 INTCPS_ISR_SET1 Register (offset = B0h) [reset = 0h] 167
2.7.18 INTCPS_ISR_CLEAR1 Register (offset = B4h) [reset = 0h] 168
2.7.19 INTCPS_PENDING_IRQ1 Register (offset = B8h) [reset = 0h] 169
2.7.20 INTCPS_PENDING_FIQ1 Register (offset = BCh) [reset = 0h] 170
2.7.21 INTCPS_ITR2 Register (offset = C0h) [reset = 0h] 171
2.7.22 INTCPS_MIR2 Register (offset = C4h) [reset = 0h] 172
2.7.23 INTCPS_SYSSTATUS Register (offset = 14h) [reset = 0h] 173
2.7.24 INTCPS_MIR_CLEAR2 Register (offset = C8h) [reset = 0h] 174
2.7.25 INTCPS_MIR_SET2 Register (offset = CCh) [reset = 0h] 175
2.7.26 INTCPS_ISR_SET2 Register (offset = D0h) [reset = 0h] 176
2.7.27 INTCPS_ISR_CLEAR2 Register (offset = D4h) [reset = 0h] 177
2.7.28 INTCPS_PENDING_IRQ2 Register (offset = D8h) [reset = 0h] 178
2.7.29 INTCPS_PENDING_FIQ2 Register (offset = DCh) [reset = 0h] 179
2.7.30 INTCPS_ITR3 Register (offset = E0h) [reset = 0h] 180
2.7.31 INTCPS_MIR3 Register (offset = E4h) [reset = 0h] 181
2.7.32 INTCPS_MIR_CLEAR3 Register (offset = E8h) [reset = 0h] 182
2.7.33 INTCPS_MIR_SET3 Register (offset = ECh) [reset = 0h] 183
2.7.34 INTCPS_ISR_SET3 Register (offset = F0h) [reset = 0h] 184
2.7.35 INTCPS_ISR_CLEAR3 Register (offset = F4h) [reset = 0h] 185

INTERNAL ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.36 INTCPS_PENDING_IRQ3 Register (offset = F8h) [reset = 0h] 186
2.7.37 INTCPS_PENDING_FIQ3 Register (offset = FCh) [reset = 0h] 187
2.7.38 INTCPS_ILR0 Register (offset = 100h) [reset = 0h] 188
2.7.39 INTCPS_ILR1 Register (offset = 104h) [reset = 0h] 189
2.7.40 INTCPS_ILR2 Register (offset = 108h) [reset = 0h] 190
2.7.41 INTCPS_ILR3 Register (offset = 10Ch) [reset = 0h] 191
2.7.42 INTCPS_ILR4 Register (offset = 110h) [reset = 0h] 192
2.7.43 INTCPS_ILR5 Register (offset = 114h) [reset = 0h] 193
2.7.44 INTCPS_ILR6 Register (offset = 118h) [reset = 0h] 194
2.7.45 INTCPS_ILR7 Register (offset = 11Ch) [reset = 0h] 195
2.7.46 INTCPS_ILR8 Register (offset = 120h) [reset = 0h] 196
2.7.47 INTCPS_ILR9 Register (offset = 124h) [reset = 0h] 197
2.7.48 INTCPS_ILR10 Register (offset = 128h) [reset = 0h] 198
2.7.49 INTCPS_ILR11 Register (offset = 12Ch) [reset = 0h] 199
2.7.50 INTCPS_ILR12 Register (offset = 130h) [reset = 0h] 200
2.7.51 INTCPS_ILR13 Register (offset = 134h) [reset = 0h] 201
2.7.52 INTCPS_ILR14 Register (offset = 138h) [reset = 0h] 202
2.7.53 INTCPS_ILR15 Register (offset = 13Ch) [reset = 0h] 203
2.7.54 INTCPS_ILR16 Register (offset = 140h) [reset = 0h] 204
2.7.55 INTCPS_ILR17 Register (offset = 144h) [reset = 0h] 205
2.7.56 INTCPS_ILR18 Register (offset = 148h) [reset = 0h] 206
2.7.57 INTCPS_ILR19 Register (offset = 14Ch) [reset = 0h] 207
2.7.58 INTCPS_ILR20 Register (offset = 150h) [reset = 0h] 208
2.7.59 INTCPS_ILR21 Register (offset = 154h) [reset = 0h] 209
2.7.60 INTCPS_ILR22 Register (offset = 158h) [reset = 0h] 210
2.7.61 INTCPS_ILR23 Register (offset = 15Ch) [reset = 0h] 211
2.7.62 INTCPS_ILR24 Register (offset = 160h) [reset = 0h] 212
2.7.63 INTCPS_ILR25 Register (offset = 164h) [reset = 0h] 213
2.7.64 INTCPS_ILR26 Register (offset = 168h) [reset = 0h] 214
2.7.65 INTCPS_ILR27 Register (offset = 16Ch) [reset = 0h] 215
2.7.66 INTCPS_ILR28 Register (offset = 170h) [reset = 0h] 216
2.7.67 INTCPS_ILR29 Register (offset = 174h) [reset = 0h] 217
2.7.68 INTCPS_ILR30 Register (offset = 178h) [reset = 0h] 218
2.7.69 INTCPS_ILR31 Register (offset = 17Ch) [reset = 0h] 219
2.7.70 INTCPS_ILR32 Register (offset = 180h) [reset = 0h] 220
2.7.71 INTCPS_ILR33 Register (offset = 184h) [reset = 0h] 221
2.7.72 INTCPS_ILR34 Register (offset = 188h) [reset = 0h] 222
2.7.73 INTCPS_ILR35 Register (offset = 18Ch) [reset = 0h] 223
2.7.74 INTCPS_ILR36 Register (offset = 190h) [reset = 0h] 224
2.7.75 INTCPS_ILR37 Register (offset = 194h) [reset = 0h] 225
2.7.76 INTCPS_ILR38 Register (offset = 198h) [reset = 0h] 226
2.7.77 INTCPS_ILR39 Register (offset = 19Ch) [reset = 0h] 227
2.7.78 INTCPS_ILR40 Register (offset = 1A0h) [reset = 0h] 228
2.7.79 INTCPS_ILR41 Register (offset = 1A4h) [reset = 0h] 229
2.7.80 INTCPS_ILR42 Register (offset = 1A8h) [reset = 0h] 230
2.7.81 INTCPS_ILR43 Register (offset = 1ACh) [reset = 0h] 231
2.7.82 INTCPS_ILR44 Register (offset = 1B0h) [reset = 0h] 232
2.7.83 INTCPS_ILR45 Register (offset = 1B4h) [reset = 0h] 233
2.7.84 INTCPS_ILR46 Register (offset = 1B8h) [reset = 0h] 234
2.7.85 INTCPS_ILR47 Register (offset = 1BCh) [reset = 0h] 235
2.7.86 INTCPS_ILR48 Register (offset = 1C0h) [reset = 0h] 236
2.7.87 INTCPS_ILR49 Register (offset = 1C4h) [reset = 0h] 237
2.7.88 INTCPS_ILR50 Register (offset = 1C8h) [reset = 0h] 238

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.89 INTCPS_ILR51 Register (offset = 1CCh) [reset = 0h] 239
2.7.90 INTCPS_ILR52 Register (offset = 1D0h) [reset = 0h] 240
2.7.91 INTCPS_ILR53 Register (offset = 1D4h) [reset = 0h] 241
2.7.92 INTCPS_ILR54 Register (offset = 1D8h) [reset = 0h] 242
2.7.93 INTCPS_ILR55 Register (offset = 1DCh) [reset = 0h] 243
2.7.94 INTCPS_ILR56 Register (offset = 1E0h) [reset = 0h] 244
2.7.95 INTCPS_ILR57 Register (offset = 1E4h) [reset = 0h] 245
2.7.96 INTCPS_ILR58 Register (offset = 1E8h) [reset = 0h] 246
2.7.97 INTCPS_ILR59 Register (offset = 1ECh) [reset = 0h] 247
2.7.98 INTCPS_ILR60 Register (offset = 1F0h) [reset = 0h] 248
2.7.99 INTCPS_ILR61 Register (offset = 1F4h) [reset = 0h] 249
2.7.100 INTCPS_ILR62 Register (offset = 1F8h) [reset = 0h] 250
2.7.101 INTCPS_ILR63 Register (offset = 1FCh) [reset = 0h] 251
2.7.102 INTCPS_ILR64 Register (offset = 200h) [reset = 0h] 252
2.7.103 INTCPS_ILR65 Register (offset = 204h) [reset = 0h] 253
2.7.104 INTCPS_ILR66 Register (offset = 208h) [reset = 0h] 254
2.7.105 INTCPS_ILR67 Register (offset = 20Ch) [reset = 0h] 255
2.7.106 INTCPS_ILR68 Register (offset = 210h) [reset = 0h] 256
2.7.107 INTCPS_ILR69 Register (offset = 214h) [reset = 0h] 257
2.7.108 INTCPS_ILR70 Register (offset = 218h) [reset = 0h] 258
2.7.109 INTCPS_ILR71 Register (offset = 21Ch) [reset = 0h] 259
2.7.110 INTCPS_ILR72 Register (offset = 220h) [reset = 0h] 260
2.7.111 INTCPS_ILR73 Register (offset = 224h) [reset = 0h] 261
2.7.112 INTCPS_ILR74 Register (offset = 228h) [reset = 0h] 262
2.7.113 INTCPS_ILR75 Register (offset = 22Ch) [reset = 0h] 263
2.7.114 INTCPS_ILR76 Register (offset = 230h) [reset = 0h] 264
2.7.115 INTCPS_ILR77 Register (offset = 234h) [reset = 0h] 265
2.7.116 INTCPS_ILR78 Register (offset = 238h) [reset = 0h] 266
2.7.117 INTCPS_ILR79 Register (offset = 23Ch) [reset = 0h] 267
2.7.118 INTCPS_ILR80 Register (offset = 240h) [reset = 0h] 268
2.7.119 INTCPS_ILR81 Register (offset = 244h) [reset = 0h] 269
2.7.120 INTCPS_ILR82 Register (offset = 248h) [reset = 0h] 270
2.7.121 INTCPS_ILR83 Register (offset = 24Ch) [reset = 0h] 271
2.7.122 INTCPS_ILR84 Register (offset = 250h) [reset = 0h] 272
2.7.123 INTCPS_ILR85 Register (offset = 254h) [reset = 0h] 273
2.7.124 INTCPS_ILR86 Register (offset = 258h) [reset = 0h] 274
2.7.125 INTCPS_ILR87 Register (offset = 25Ch) [reset = 0h] 275
2.7.126 INTCPS_ILR88 Register (offset = 260h) [reset = 0h] 276
2.7.127 INTCPS_ILR89 Register (offset = 264h) [reset = 0h] 277
2.7.128 INTCPS_ILR90 Register (offset = 268h) [reset = 0h] 278
2.7.129 INTCPS_ILR91 Register (offset = 26Ch) [reset = 0h] 279
2.7.130 INTCPS_ILR92 Register (offset = 270h) [reset = 0h] 280
2.7.131 INTCPS_ILR93 Register (offset = 274h) [reset = 0h] 281
2.7.132 INTCPS_ILR94 Register (offset = 278h) [reset = 0h] 282
2.7.133 INTCPS_ILR95 Register (offset = 27Ch) [reset = 0h] 283
2.7.134 INTCPS_SIR_IRQ Register (offset = 40h) [reset = 0h] 284
2.7.135 INTCPS_ILR96 Register (offset = 280h) [reset = 0h] 285
2.7.136 INTCPS_ILR97 Register (offset = 284h) [reset = 0h] 286
2.7.137 INTCPS_ILR98 Register (offset = 288h) [reset = 0h] 287
2.7.138 INTCPS_ILR99 Register (offset = 28Ch) [reset = 0h] 288
2.7.139 INTCPS_ILR100 Register (offset = 290h) [reset = 0h] 289
2.7.140 INTCPS_ILR101 Register (offset = 294h) [reset = 0h] 290
2.7.141 INTCPS_ILR102 Register (offset = 298h) [reset = 0h] 291

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.142 INTCPS_ILR103 Register (offset = 29Ch) [reset = 0h] 292
2.7.143 INTCPS_ILR104 Register (offset = 2A0h) [reset = 0h] 293
2.7.144 INTCPS_ILR105 Register (offset = 2A4h) [reset = 0h] 294
2.7.145 INTCPS_SIR_FIQ Register (offset = 44h) [reset = 0h] 295
2.7.146 INTCPS_ILR106 Register (offset = 2A8h) [reset = 0h] 296
2.7.147 INTCPS_ILR107 Register (offset = 2ACh) [reset = 0h] 297
2.7.148 INTCPS_ILR108 Register (offset = 2B0h) [reset = 0h] 298
2.7.149 INTCPS_ILR109 Register (offset = 2B4h) [reset = 0h] 299
2.7.150 INTCPS_ILR110 Register (offset = 2B8h) [reset = 0h] 300
2.7.151 INTCPS_ILR111 Register (offset = 2BCh) [reset = 0h] 301
2.7.152 INTCPS_ILR112 Register (offset = 2C0h) [reset = 0h] 302
2.7.153 INTCPS_ILR113 Register (offset = 2C4h) [reset = 0h] 303
2.7.154 INTCPS_ILR114 Register (offset = 2C8h) [reset = 0h] 304
2.7.155 INTCPS_ILR115 Register (offset = 2CCh) [reset = 0h] 305
2.7.156 INTCPS_CONTROL Register (offset = 48h) [reset = 0h] 306
2.7.157 INTCPS_ILR116 Register (offset = 2D0h) [reset = 0h] 307
2.7.158 INTCPS_ILR117 Register (offset = 2D4h) [reset = 0h] 308
2.7.159 INTCPS_ILR118 Register (offset = 2D8h) [reset = 0h] 309
2.7.160 INTCPS_ILR119 Register (offset = 2DCh) [reset = 0h] 310
2.7.161 INTCPS_ILR120 Register (offset = 2E0h) [reset = 0h] 311
2.7.162 INTCPS_ILR121 Register (offset = 2E4h) [reset = 0h] 312
2.7.163 INTCPS_ILR122 Register (offset = 2E8h) [reset = 0h] 313
2.7.164 INTCPS_ILR123 Register (offset = 2ECh) [reset = 0h] 314
2.7.165 INTCPS_ILR124 Register (offset = 2F0h) [reset = 0h] 315
2.7.166 INTCPS_ILR125 Register (offset = 2F4h) [reset = 0h] 316
2.7.167 INTCPS_PROTECTION Register (offset = 4Ch) [reset = 0h] 317
2.7.168 INTCPS_ILR126 Register (offset = 2F8h) [reset = 0h] 318
2.7.169 INTCPS_ILR127 Register (offset = 2FCh) [reset = 0h] 319
2.7.170 INTCPS_IDLE Register (offset = 50h) [reset = 0h] 320
2.7.171 INTCPS_IRQ_PRIORITY Register (offset = 60h) [reset = 0h] 321

3 Memory Map 323
3.1 ARM Cortex A8 Memory Map 323
3.2 ARM Cortex M3 Memory Map 326
3.3 L4 Memory Map 328

4 Programmable Real-Time Unit Subsystem (PRUSS) 337
4.1 Introduction 338
4.1.1 Features 338
4.2 Integration 339
4.3 Functional Description 339
4.4 Use Cases 339
4.5 PRUSS_INTC Registers 339
4.5.1 REVID Register (offset = 0h) [reset = 4E82A900h] 342
4.5.2 CMR0 Register (offset = 400h) [reset = 0h] 343
4.5.3 CMR1 Register (offset = 404h) [reset = 0h] 344
4.5.4 CMR2 Register (offset = 408h) [reset = 0h] 345
4.5.5 CMR3 Register (offset = 40Ch) [reset = 0h] 346
4.5.6 CMR4 Register (offset = 410h) [reset = 0h] 347
4.5.7 CMR5 Register (offset = 414h) [reset = 0h] 348
4.5.8 CMR6 Register (offset = 418h) [reset = 0h] 349
4.5.9 CMR7 Register (offset = 41Ch) [reset = 0h] 350
4.5.10 CMR8 Register (offset = 420h) [reset = 0h] 351
4.5.11 CMR9 Register (offset = 424h) [reset = 0h] 352
4.5.12 CMR10 Register (offset = 428h) [reset = 0h] 353

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.13 CMR11 Register (offset = 42Ch) [reset = 0h] 354
4.5.14 CMR12 Register (offset = 430h) [reset = 0h] 355
4.5.15 CMR13 Register (offset = 434h) [reset = 0h] 356
4.5.16 CMR14 Register (offset = 438h) [reset = 0h] 357
4.5.17 CMR15 Register (offset = 43Ch) [reset = 0h] 358
4.5.18 GPIR Register (offset = 80h) [reset = 80000000h] 359
4.5.19 GER Register (offset = 10h) [reset = 0h] 360
4.5.20 HMR0 Register (offset = 800h) [reset = 0h] 361
4.5.21 HMR1 Register (offset = 804h) [reset = 0h] 362
4.5.22 HMR2 Register (offset = 808h) [reset = 0h] 363
4.5.23 HIPIR0 Register (offset = 900h) [reset = 80000000h] 364
4.5.24 HIPIR1 Register (offset = 904h) [reset = 80000000h] 365
4.5.25 HIPIR2 Register (offset = 908h) [reset = 80000000h] 366
4.5.26 HIPIR3 Register (offset = 90Ch) [reset = 80000000h] 367
4.5.27 HIPIR4 Register (offset = 910h) [reset = 80000000h] 368
4.5.28 HIPIR5 Register (offset = 914h) [reset = 80000000h] 369
4.5.29 HIPIR6 Register (offset = 918h) [reset = 80000000h] 370
4.5.30 HIPIR7 Register (offset = 91Ch) [reset = 80000000h] 371
4.5.31 HIPIR8 Register (offset = 920h) [reset = 80000000h] 372
4.5.32 HIPIR9 Register (offset = 924h) [reset = 80000000h] 373
4.5.33 GNLR Register (offset = 1Ch) [reset = 100h] 374
4.5.34 SISR Register (offset = 20h) [reset = 0h] 375
4.5.35 SIPR0 Register (offset = D00h) [reset = 1h] 376
4.5.36 SIPR1 Register (offset = D04h) [reset = 1h] 377
4.5.37 SITR0 Register (offset = D80h) [reset = 0h] 378
4.5.38 SITR1 Register (offset = D84h) [reset = 0h] 379
4.5.39 SICR Register (offset = 24h) [reset = 0h] 380
4.5.40 CR Register (offset = 4h) [reset = 0h] 381
4.5.41 EISR Register (offset = 28h) [reset = 0h] 382
4.5.42 HINLR0 Register (offset = 1100h) [reset = 100h] 383
4.5.43 HINLR1 Register (offset = 1104h) [reset = 100h] 384
4.5.44 HINLR2 Register (offset = 1108h) [reset = 100h] 385
4.5.45 HINLR3 Register (offset = 110Ch) [reset = 100h] 386
4.5.46 HINLR4 Register (offset = 1110h) [reset = 100h] 387
4.5.47 HINLR5 Register (offset = 1114h) [reset = 100h] 388
4.5.48 HINLR6 Register (offset = 1118h) [reset = 100h] 389
4.5.49 HINLR7 Register (offset = 111Ch) [reset = 100h] 390
4.5.50 HINLR8 Register (offset = 1120h) [reset = 100h] 391
4.5.51 HINLR9 Register (offset = 1124h) [reset = 100h] 392
4.5.52 EICR Register (offset = 2Ch) [reset = 0h] 393
4.5.53 SRSR0 Register (offset = 200h) [reset = 0h] 394
4.5.54 SRSR1 Register (offset = 204h) [reset = 0h] 395
4.5.55 HIEISR Register (offset = 34h) [reset = 0h] 396
4.5.56 HIER Register (offset = 1500h) [reset = 0h] 397
4.5.57 HIDISR Register (offset = 38h) [reset = 0h] 398
4.5.58 SECR0 Register (offset = 280h) [reset = 0h] 399
4.5.59 SECR1 Register (offset = 284h) [reset = 0h] 400
4.5.60 ESR0 Register (offset = 300h) [reset = 0h] 401
4.5.61 ERS1 Register (offset = 304h) [reset = 0h] 402
4.5.62 ECR0 Register (offset = 380h) [reset = 0h] 403
4.5.63 ECR1 Register (offset = 384h) [reset = 0h] 404
4.6 PRUSS_PRU_CTRL Registers 404
4.6.1 CONTROL Register (offset = 0h) [reset = 1h] 405

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.6.2 CYCLE Register (offset = Ch) [reset = 0h] 406
4.6.3 STALL Register (offset = 10h) [reset = 0h] 407
4.6.4 CTBIR0 Register (offset = 20h) [reset = 0h] 408
4.6.5 CTBIR1 Register (offset = 24h) [reset = 0h] 409
4.6.6 STATUS Register (offset = 4h) [reset = 0h] 410
4.6.7 CTPPR0 Register (offset = 28h) [reset = 0h] 411
4.6.8 CTPPR1 Register (offset = 2Ch) [reset = 0h] 412
4.6.9 WAKEUP_EN Register (offset = 8h) [reset = 0h] 413
4.7 PRUSS_PRU_DEBUG Registers 413
4.7.1 GPR0 Register (offset = 0h) [reset = 0h] 415
4.7.2 GPR25 Register (offset = 64h) [reset = 0h] 416
4.7.3 GPR26 Register (offset = 68h) [reset = 0h] 417
4.7.4 GPR27 Register (offset = 6Ch) [reset = 0h] 418
4.7.5 GPR28 Register (offset = 70h) [reset = 0h] 419
4.7.6 GPR29 Register (offset = 74h) [reset = 0h] 420
4.7.7 GPR3 Register (offset = Ch) [reset = 0h] 421
4.7.8 GPR30 Register (offset = 78h) [reset = 0h] 422
4.7.9 GPR31 Register (offset = 7Ch) [reset = 0h] 423
4.7.10 CTR0 Register (offset = 80h) [reset = 20000h] 424
4.7.11 CTR1 Register (offset = 84h) [reset = 48040000h] 425
4.7.12 CTR2 Register (offset = 88h) [reset = 4802A000h] 426
4.7.13 CTR3 Register (offset = 8Ch) [reset = 30000h] 427
4.7.14 CTR4 Register (offset = 90h) [reset = 26000h] 428
4.7.15 CTR5 Register (offset = 94h) [reset = 48060000h] 429
4.7.16 CTR6 Register (offset = 98h) [reset = 48030000h] 430
4.7.17 CTR7 Register (offset = 9Ch) [reset = 28000h] 431
4.7.18 GPR4 Register (offset = 10h) [reset = 0h] 432
4.7.19 CTR8 Register (offset = A0h) [reset = 46000000h] 433
4.7.20 CTR9 Register (offset = A4h) [reset = 4A100000h] 434
4.7.21 CTR10 Register (offset = A8h) [reset = 48318000h] 435
4.7.22 CTR11 Register (offset = ACh) [reset = 48022000h] 436
4.7.23 CTR12 Register (offset = B0h) [reset = 48024000h] 437
4.7.24 CTR13 Register (offset = B4h) [reset = 48310000h] 438
4.7.25 CTR14 Register (offset = B8h) [reset = 481CC000h] 439
4.7.26 CTR15 Register (offset = BCh) [reset = 481D0000h] 440
4.7.27 CTR16 Register (offset = C0h) [reset = 481A0000h] 441
4.7.28 CTR17 Register (offset = C4h) [reset = 4819C000h] 442
4.7.29 GPR5 Register (offset = 14h) [reset = 0h] 443
4.7.30 CTR18 Register (offset = C8h) [reset = 48300000h] 444
4.7.31 CTR19 Register (offset = CCh) [reset = 48302000h] 445
4.7.32 CTR20 Register (offset = D0h) [reset = 48304000h] 446
4.7.33 CTR21 Register (offset = D4h) [reset = 32400h] 447
4.7.34 CTR22 Register (offset = D8h) [reset = 480C8000h] 448
4.7.35 CTR23 Register (offset = DCh) [reset = 480CA000h] 449
4.7.36 CTR24 Register (offset = E0h) [reset = 0h] 450
4.7.37 GPR6 Register (offset = 18h) [reset = 0h] 451
4.7.38 GPR7 Register (offset = 1Ch) [reset = 0h] 452
4.7.39 GPR8 Register (offset = 20h) [reset = 0h] 453
4.7.40 GPR9 Register (offset = 24h) [reset = 0h] 454
4.7.41 GPR1 Register (offset = 4h) [reset = 0h] 455
4.7.42 GPR10 Register (offset = 28h) [reset = 0h] 456
4.7.43 GPR11 Register (offset = 2Ch) [reset = 0h] 457
4.7.44 GPR12 Register (offset = 30h) [reset = 0h] 458

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.45 GPR13 Register (offset = 34h) [reset = 0h] 459
4.7.46 GPR14 Register (offset = 38h) [reset = 0h] 460
4.7.47 GPR15 Register (offset = 3Ch) [reset = 0h] 461
4.7.48 GPR16 Register (offset = 40h) [reset = 0h] 462
4.7.49 GPR17 Register (offset = 44h) [reset = 0h] 463
4.7.50 GPR18 Register (offset = 48h) [reset = 0h] 464
4.7.51 GPR19 Register (offset = 4Ch) [reset = 0h] 465
4.7.52 GPR2 Register (offset = 8h) [reset = 0h] 466
4.7.53 GPR20 Register (offset = 50h) [reset = 0h] 467
4.7.54 GPR21 Register (offset = 54h) [reset = 0h] 468
4.7.55 GPR22 Register (offset = 58h) [reset = 0h] 469
4.7.56 GPR23 Register (offset = 5Ch) [reset = 0h] 470
4.7.57 GPR24 Register (offset = 60h) [reset = 0h] 471
4.8 PRUSS_CFG Registers 471
4.8.1 REVID Register (offset = 0h) [reset = 47000000h] 472
4.8.2 GPCFG1 Register (offset = Ch) [reset = 0h] 473
4.8.3 CGR Register (offset = 10h) [reset = 24924h] 474
4.8.4 ISRP Register (offset = 14h) [reset = 0h] 475
4.8.5 ISP Register (offset = 18h) [reset = 0h] 476
4.8.6 IESP Register (offset = 1Ch) [reset = 0h] 477
4.8.7 IECP Register (offset = 20h) [reset = 0h] 478
4.8.8 SCRP Register (offset = 24h) [reset = AA84Bh] 479
4.8.9 SYSCFG Register (offset = 4h) [reset = 1Ah] 480
4.8.10 PMAO Register (offset = 28h) [reset = 0h] 481
4.8.11 MII_RT Register (offset = 2Ch) [reset = 1h] 482
4.8.12 IEPCLK Register (offset = 30h) [reset = 0h] 483
4.8.13 SPP Register (offset = 34h) [reset = 0h] 484
4.8.14 PIN_MX Register (offset = 40h) [reset = 0h] 485
4.8.15 GPCFG0 Register (offset = 8h) [reset = 0h] 486
4.9 PRUSS_UART Registers 486
4.9.1 UART RBR/TBR Registers Register (offset = 0h) [reset = 0h] 488
4.9.2 UART Line Control Register Register (offset = Ch) [reset = 0h] 489
4.9.3 UART Modem Control Register Register (offset = 10h) [reset = 0h] 490
4.9.4 UART Line Status Register Register (offset = 14h) [reset = 60h] 491
4.9.5 UART Modem Status Register Register (offset = 18h) [reset = 0h] 492
4.9.6 UART Scratch Register Register (offset = 1Ch) [reset = 0h] 493
4.9.7 UART Divisor Register (LSB) Register (offset = 20h) [reset = 0h] 494
4.9.8 UART Divisor Register (MSB) Register (offset = 24h) [reset = 0h] 495
4.9.9 UART Interrupt Enable Register Register (offset = 4h) [reset = 0h] 496
4.9.10 UART Peripheral ID Register Register (offset = 28h) [reset = 44141102h] 497
4.9.11 UART PowerManagement and Emulation Register Register (offset = 30h) [reset = 0h] 498
4.9.12 UART Mode Definition Register Register (offset = 34h) [reset = 0h] 499
4.9.13 UART Interrupt Identification Register / FIFO Control Register Register (offset = 8h) [reset = 1h] . 500
4.10 PRUSS_IEP Registers 500
4.10.1 GLOBAL_CFG Register (offset = 0h) [reset = 550h] 502
4.10.2 CMP7 Register (offset = 64h) [reset = 0h] 503
4.10.3 COUNT Register (offset = Ch) [reset = 0h] 504
4.10.4 RXIPG0 Register (offset = 80h) [reset = FFFF0000h] 505
4.10.5 RXIPG1 Register (offset = 84h) [reset = FFFF0000h] 506
4.10.6 CAP_CFG Register (offset = 10h) [reset = 1FC00h] 507
4.10.7 CAP_STATUS Register (offset = 14h) [reset = 0h] 508
4.10.8 CAPR0 Register (offset = 18h) [reset = 0h] 509
4.10.9 SYNC_CTRL Register (offset = 100h) [reset = 0h] 510

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.10 SYNC_FIRST_STAT Register (offset = 104h) [reset = 0h] 511
4.10.11 SYNC0_STAT Register (offset = 108h) [reset = 0h] 512
4.10.12 SYNC1_STAT Register (offset = 10Ch) [reset = 0h] 513
4.10.13 SYNC_PWIDTH Register (offset = 110h) [reset = 0h] 514
4.10.14 SYNC0_PERIOD Register (offset = 114h) [reset = 1h] 515
4.10.15 CAPR1 Register (offset = 1Ch) [reset = 0h] 516
4.10.16 SYNC1_DELAY Register (offset = 118h) [reset = 0h] 517
4.10.17 SYNC_START Register (offset = 11Ch) [reset = 0h] 518
4.10.18 CAPR2 Register (offset = 20h) [reset = 0h] 519
4.10.19 CAPR3 Register (offset = 24h) [reset = 0h] 520
4.10.20 GLOBAL_STATUS Register (offset = 4h) [reset = 0h] 521
4.10.21 CAPR4 Register (offset = 28h) [reset = 0h] 522
4.10.22 CAPR5 Register (offset = 2Ch) [reset = 0h] 523
4.10.23 CAPR6 Register (offset = 30h) [reset = 0h] 524
4.10.24 WD_PREDIV Register (offset = 200h) [reset = 4E20h] 525
4.10.25 PDI_WD_TIM Register (offset = 204h) [reset = 3E8h] 526
4.10.26 CAPF6 Register (offset = 34h) [reset = 0h] 527
4.10.27 PD_WD_TIM Register (offset = 208h) [reset = 3E8h] 528
4.10.28 WD_STATUS Register (offset = 20Ch) [reset = 10001h] 529
4.10.29 WD_EXP_CNT Register (offset = 210h) [reset = 0h] 530
4.10.30 WD_CTRL Register (offset = 214h) [reset = 0h] 531
4.10.31 CAPR7 Register (offset = 38h) [reset = 0h] 532
4.10.32 CAPF7 Register (offset = 3Ch) [reset = 0h] 533
4.10.33 CMP_CFG Register (offset = 40h) [reset = 0h] 534
4.10.34 CMP_STATUS Register (offset = 44h) [reset = 0h] 535
4.10.35 CMP0 Register (offset = 48h) [reset = 0h] 536
4.10.36 CMP1 Register (offset = 4Ch) [reset = 0h] 537
4.10.37 DIGIO_CTRL Register (offset = 300h) [reset = 4h] 538
4.10.38 DIGIO_STATUS Register (offset = 304h) [reset = 0h] 539
4.10.39 DIGIO_DATA_IN Register (offset = 308h) [reset = 0h] 540
4.10.40 DIGIO_DATA_IN_RAW Register (offset = 30Ch) [reset = 0h] 541
4.10.41 DIGIO_DATA_OUT Register (offset = 310h) [reset = 0h] 542
4.10.42 DIGIO_DATA_OUT_EN Register (offset = 314h) [reset = 0h] 543
4.10.43 DIGIO_EXP Register (offset = 318h) [reset = 20h] 544
4.10.44 COMPEN Register (offset = 8h) [reset = 0h] 545
4.10.45 CMP2 Register (offset = 50h) [reset = 0h] 546
4.10.46 CMP3 Register (offset = 54h) [reset = 0h] 547
4.10.47 CMP4 Register (offset = 58h) [reset = 0h] 548
4.10.48 CMP5 Register (offset = 5Ch) [reset = 0h] 549
4.10.49 CMP6 Register (offset = 60h) [reset = 0h] 550
4.11 PRUSS_ECAP Registers 550
4.11.1 TSCNT Register (offset = 0h) [reset = 0h] 551
4.11.2 CAP2 Register (offset = Ch) [reset = 0h] 552
4.11.3 CAP3 Register (offset = 10h) [reset = 0h] 553
4.11.4 CAP4 Register (offset = 14h) [reset = 0h] 554
4.11.5 ECCTL1 Register (offset = 26h) [reset = 0h] 555
4.11.6 CNTPHS Register (offset = 4h) [reset = 0h] 557
4.11.7 ECCTL2 Register (offset = 28h) [reset = 6h] 558
4.11.8 ECEINT Register (offset = 2Ah) [reset = 0h] 560
4.11.9 ECFLG Register (offset = 2Ch) [reset = 0h] 561
4.11.10 ECCLR Register (offset = 30h) [reset = 0h] 562
4.11.11 ECFRC Register (offset = 34h) [reset = 0h] 563
4.11.12 CAP1 Register (offset = 8h) [reset = 0h] 564

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.11.13 PID Register (offset = 5Ch) [reset = 44D22100h] ... 565
4.12 PRUSS_MII_RT_CFG Registers ... 565
4.12.1 RXCFG0 Register (offset = 0h) [reset = 0h] ... 566
4.12.2 TXCFG0 Register (offset = 10h) [reset = 400100h] ... 567
4.12.3 TXCFG1 Register (offset = 14h) [reset = 400000h] ... 568
4.12.4 TX_CRC0 Register (offset = 20h) [reset = 0h] ... 569
4.12.5 TX_CRC1 Register (offset = 24h) [reset = 0h] ... 570
4.12.6 RXCFG1 Register (offset = 4h) [reset = 8h] ... 571
4.12.7 TX_IPG0 Register (offset = 30h) [reset = 28h] ... 572
4.12.8 TX_IPG1 Register (offset = 34h) [reset = 28h] ... 573
4.12.9 PRS0 Register (offset = 38h) [reset = 0h] ... 574
4.12.10 PRS1 Register (offset = 3Ch) [reset = 0h] ... 575
4.12.11 RX_FRMS0 Register (offset = 40h) [reset = 5F1003Fh] ... 576
4.12.12 RX_FRMS1 Register (offset = 44h) [reset = 5F1003Fh] ... 577
4.12.13 RX_PCNT0 Register (offset = 48h) [reset = E1h] ... 578
4.12.14 RX_PCNT1 Register (offset = 4Ch) [reset = E1h] ... 579
4.12.15 RX_ERR0 Register (offset = 50h) [reset = 0h] ... 580
4.12.16 RX_ERR1 Register (offset = 54h) [reset = 0h] ... 581
4.13 PRUSS_MII_MDIO Registers ... 581
4.13.1 VER Register (offset = 0h) [reset = 70105h] ... 583
4.13.2 LINK Register (offset = Ch) [reset = 0h] ... 584
4.13.3 USERACCESS0 Register (offset = 80h) [reset = 0h] ... 585
4.13.4 USERPHYSEL0 Register (offset = 84h) [reset = 0h] ... 586
4.13.5 USERACCESS1 Register (offset = 88h) [reset = 0h] ... 587
4.13.6 USERPHYSEL1 Register (offset = 8Ch) [reset = 0h] ... 588
4.13.7 LINKINTRAW Register (offset = 10h) [reset = 0h] ... 589
4.13.8 LINKINTMASKED Register (offset = 14h) [reset = 0h] ... 590
4.13.9 USERINTRAW Register (offset = 20h) [reset = 0h] ... 591
4.13.10 USERINTMASKED Register (offset = 24h) [reset = 0h] ... 592
4.13.11 CONTROL Register (offset = 4h) [reset = 810000FFh] ... 593
4.13.12 USERINTMASKSET Register (offset = 28h) [reset = 0h] ... 594
4.13.13 USERINTMASKCLR Register (offset = 2Ch) [reset = 0h] ... 595
4.13.14 ALIVE Register (offset = 8h) [reset = 0h] ... 596

5 Graphics Accelerator (SGX) ... 597
5.1 Introduction ... 598
5.2 Integration ... 598
5.3 Functional Description ... 598
5.4 Use Cases ... 598
5.5 Registers ... 599
5.5.1 SGX530_MMAV10GS70 Registers ... 599

6 Memory Subsystem ... 847
6.1 GPMC ... 848
6.1.1 Introduction ... 848
6.1.2 Integration ... 852
6.1.3 Functional Description ... 854
6.1.4 Use Cases ... 955
6.1.5 GPMC Registers ... 967
6.2 OCMC-RAM ... 1000
6.2.1 Introduction ... 1000
6.2.2 Integration ... 1001
6.2.3 Functional Description ... 1001
6.2.4 Use Cases ... 1001
6.2.5 Registers ... 1002

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3 EMIF 1003
6.3.1 Introduction 1003
6.3.2 Integration 1005
6.3.3 Functional Description 1006
6.3.4 Use Cases 1006
6.3.5 EMIF4D Registers 1007
6.4 ELM 1059
6.4.1 Introduction 1059
6.4.2 Integration 1060
6.4.3 Functional Description 1061
6.4.4 Basic Programming Model 1064
6.4.5 ELM Registers 1070

7 Power and Clock Management (PRCM) 1083
7.1 Power Management 1084
7.1.1 Introduction 1084
7.1.2 Integration 1084
7.1.3 Functional Description 1084
7.1.4 Use Cases 1084
7.1.5 Registers 1085
7.2 Clock Management 1113
7.2.1 Introduction 1113
7.2.2 Integration 1113
7.2.3 Functional Description 1113
7.2.4 Use Cases 1113
7.2.5 Registers 1114

8 Control Module 1293

9 Interconnects 1295
9.1 Introduction 1296
9.2 Integration 1296
9.3 Functional Description 1296
9.4 Use Cases 1296
9.5 Interconnect Registers 1297
9.5.1 L3 Interconnect Registers 1297
9.5.2 L4 Interconnects Registers 1329

10 Enhanced Direct Memory Access (EDMA) 1359
10.1 Introduction 1360
10.2 Integration 1360
10.3 Functional Description 1360
10.4 Use Cases 1360
10.5 EDMA3 Registers 1361
10.5.1 Register Memory Maps 1361
10.5.2 EDMA3 Channel Controller Control Registers 1361
10.5.3 EDMA3 Transfer Controller Control Registers 1414

11 Touchscreen Controller 1439
11.1 Introduction 1440
11.1.1 TSC_ADC Features 1440
11.1.2 Unsupported TSC_ADC_SS Features 1440
11.2 Integration 1441
11.2.1 TSC_ADC Connectivity Attributes 1441
11.2.2 TSC_ADC Clock and Reset Management 1442
11.2.3 TSC_ADC Pin List 1442
11.3 Functional Description 1443

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.3.1 HW Synchronized or SW Channels 1443
11.3.2 Open Delay and Sample Delay 1443
11.3.3 Averaging of Samples (1, 2, 4, 8, and 16) 1443
11.3.4 One-Shot (Single) or Continuous Mode 1443
11.3.5 Interrupts 1443
11.3.6 DMA Requests 1443
11.3.7 AFE Functional Block Diagram 1444
11.4 Operational Modes 1446
11.4.1 PenCtrl and PenIRQ 1447
11.5 TSC_ADC_SS Registers 1447
11.5.1 REVISION Register (offset = 0h) [reset = 47300001h] 1450
11.5.2 STEPCONFIG4 Register (offset = 64h) [reset = 0h] 1451
11.5.3 STEPDELAY4 Register (offset = 68h) [reset = 0h] 1452
11.5.4 STEPCONFIG5 Register (offset = 6Ch) [reset = 0h] 1453
11.5.5 STEPDELAY5 Register (offset = 70h) [reset = 0h] 1454
11.5.6 STEPCONFIG6 Register (offset = 74h) [reset = 0h] 1455
11.5.7 IRQSTATUS_RAW Register (offset = Ch) [reset = 0h] 1456
11.5.8 STEPDELAY6 Register (offset = 78h) [reset = 0h] 1458
11.5.9 STEPCONFIG7 Register (offset = 7Ch) [reset = 0h] 1459
11.5.10 STEPDELAY7 Register (offset = 80h) [reset = 0h] 1460
11.5.11 STEPCONFIG8 Register (offset = 84h) [reset = 0h] 1461
11.5.12 STEPDELAY8 Register (offset = 88h) [reset = 0h] 1462
11.5.13 STEPCONFIG9 Register (offset = 8Ch) [reset = 0h] 1463
11.5.14 STEPDELAY9 Register (offset = 90h) [reset = 0h] 1464
11.5.15 STEPCONFIG10 Register (offset = 94h) [reset = 0h] 1465
11.5.16 STEPDELAY10 Register (offset = 98h) [reset = 0h] 1466
11.5.17 STEPCONFIG11 Register (offset = 9Ch) [reset = 0h] 1467
11.5.18 IRQSTATUS Register (offset = 10h) [reset = 0h] 1468
11.5.19 STEPDELAY11 Register (offset = A0h) [reset = 0h] 1470
11.5.20 STEPCONFIG12 Register (offset = A4h) [reset = 0h] 1471
11.5.21 STEPDELAY12 Register (offset = A8h) [reset = 0h] 1472
11.5.22 STEPCONFIG13 Register (offset = ACh) [reset = 0h] 1473
11.5.23 STEPDELAY13 Register (offset = B0h) [reset = 0h] 1474
11.5.24 STEPCONFIG14 Register (offset = B4h) [reset = 0h] 1475
11.5.25 STEPDELAY14 Register (offset = B8h) [reset = 0h] 1476
11.5.26 STEPCONFIG15 Register (offset = BCh) [reset = 0h] 1477
11.5.27 STEPDELAY15 Register (offset = C0h) [reset = 0h] 1478
11.5.28 STEPCONFIG16 Register (offset = C4h) [reset = 0h] 1479
11.5.29 IRQENABLE_SET Register (offset = 14h) [reset = 0h] 1480
11.5.30 STEPDELAY16 Register (offset = C8h) [reset = 0h] 1482
11.5.31 FIFO0COUNT Register (offset = CCh) [reset = 0h] 1483
11.5.32 FIFO0THRESHOLD Register (offset = D0h) [reset = 0h] 1484
11.5.33 DMA0REQ Register (offset = D4h) [reset = 0h] 1485
11.5.34 FIFO1COUNT Register (offset = D8h) [reset = 0h] 1486
11.5.35 FIFO1THRESHOLD Register (offset = DCh) [reset = 0h] 1487
11.5.36 DMA1REQ Register (offset = E0h) [reset = 0h] 1488
11.5.37 FIFO0DATA Register (offset = E4h) [reset = 0h] 1489
11.5.38 FIFO1DATA Register (offset = E8h) [reset = 0h] 1490
11.5.39 IRQENABLE_CLR Register (offset = 18h) [reset = 0h] 1491
11.5.40 IRQWAKEUP Register (offset = 1Ch) [reset = 0h] 1493
11.5.41 DMAENABLE_SET Register (offset = 20h) [reset = 0h] 1494
11.5.42 DMAENABLE_CLR Register (offset = 24h) [reset = 0h] 1495
11.5.43 SYSCONFIG Register (offset = 4h) [reset = 8h] 1496

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.44 CTRL Register (offset = 28h) [reset = 0h] 1497
11.5.45 ADCSTAT Register (offset = 2Ch) [reset = 10h] 1498
11.5.46 ADCRANGE Register (offset = 30h) [reset = 0h] 1499
11.5.47 ADC_CLKDIV Register (offset = 34h) [reset = 0h] 1500
11.5.48 ADC_MISC Register (offset = 38h) [reset = 0h] 1501
11.5.49 STEPENABLE Register (offset = 3Ch) [reset = 0h] 1502
11.5.50 IDLECONFIG Register (offset = 40h) [reset = 0h] 1503
11.5.51 TS_CHARGE_STEPCONFIG Register (offset = 44h) [reset = 0h] 1504
11.5.52 TS_CHARGE_DELAY Register (offset = 48h) [reset = 1h] 1505
11.5.53 STEPCONFIG1 Register (offset = 4Ch) [reset = 0h] 1506
11.5.54 IRQ_EOI Register (offset = 8h) [reset = 0h] 1507
11.5.55 STEPDELAY1 Register (offset = 50h) [reset = 0h] 1508
11.5.56 STEPCONFIG2 Register (offset = 54h) [reset = 0h] 1509
11.5.57 STEPDELAY2 Register (offset = 58h) [reset = 0h] 1510
11.5.58 STEPCONFIG3 Register (offset = 5Ch) [reset = 0h] 1511
11.5.59 STEPDELAY3 Register (offset = 60h) [reset = 0h] 1512

12 LCD Controller 1513

12.1 Introduction 1514
12.1.1 Purpose of the Peripheral 1514
12.1.2 Features 1515
12.1.3 Unsupported Features 1515
12.2 Integration 1516
12.2.1 LCD Controller Connectivity Attributes 1516
12.2.2 LCD Controller Clock and Reset Management 1517
12.2.3 LCD Controller Pin List 1517
12.3 Functional Description 1518
12.3.1 Clocking 1518
12.3.2 LCD External I/O Signals 1520
12.3.3 DMA Engine 1521
12.3.4 LIDD Controller 1522
12.3.5 Raster Controller 1527
12.4 Use Cases 1539
12.5 LCD Registers 1539
12.5.1 pid Register (offset = 0h) [reset = 4F200000h] 1540
12.5.2 IRQENABLE_CLEAR Register (offset = 64h) [reset = 0h] 1541
12.5.3 IRQEOI_VECTOR Register (offset = 68h) [reset = 0h] 1543
12.5.4 CLKC_ENABLE Register (offset = 6Ch) [reset = 0h] 1544
12.5.5 CLKC_RESET Register (offset = 70h) [reset = 0h] 1545
12.5.6 LIDD_CTRL Register (offset = Ch) [reset = 0h] 1546
12.5.7 LIDD_CS0_CONF Register (offset = 10h) [reset = 440044h] 1548
12.5.8 LIDD_CS0_ADDR Register (offset = 14h) [reset = 0h] 1549
12.5.9 reg29 Register (offset = E8h) [reset = A00h] 1550
12.5.10 reg30 Register (offset = ECh) [reset = 0h] 1551
12.5.11 LIDD_CS0_DATA Register (offset = 18h) [reset = 0h] 1552
12.5.12 reg31 Register (offset = F0h) [reset = 0h] 1553
12.5.13 LIDD_CS1_CONF Register (offset = 1Ch) [reset = 420044h] 1554
12.5.14 LIDD_CS1_ADDR Register (offset = 20h) [reset = 0h] 1555
12.5.15 LIDD_CS1_DATA Register (offset = 24h) [reset = 0h] 1556
12.5.16 CTRL Register (offset = 4h) [reset = 0h] 1557
12.5.17 RASTER_CTRL Register (offset = 28h) [reset = 0h] 1558
12.5.18 RASTER_TIMING_0 Register (offset = 2Ch) [reset = 0h] 1560
12.5.19 RASTER_TIMING_1 Register (offset = 30h) [reset = 0h] 1561
12.5.20 RASTER_TIMING_2 Register (offset = 34h) [reset = 0h] 1562

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.21 RASTER_SUBPANEL Register (offset = 38h) [reset = 0h] 1564
12.5.22 RASTER_SUBPANEL2 Register (offset = 3Ch) [reset = 0h] 1565
12.5.23 LCDDMA_CTRL Register (offset = 40h) [reset = 0h] 1566
12.5.24 LCDDMA_FB0_BASE Register (offset = 44h) [reset = 0h] 1567
12.5.25 LCDDMA_FB0_CEILING Register (offset = 48h) [reset = 0h] 1568
12.5.26 LCDDMA_FB1_BASE Register (offset = 4Ch) [reset = 0h] 1569
12.5.27 LCDDMA_FB1_CEILING Register (offset = 50h) [reset = 0h] 1570
12.5.28 SYSCONFIG Register (offset = 54h) [reset = 28h] 1571
12.5.29 IRQSTATUS_RAW Register (offset = 58h) [reset = 0h] 1572
12.5.30 IRQSTATUS Register (offset = 5Ch) [reset = 0h] 1573
12.5.31 IRQENABLE_SET Register (offset = 60h) [reset = 0h] 1574

13 Ethernet Subsystem 1577

13.1 Introduction 1578
13.1.1 Features 1578
13.1.2 Unsupported Features 1579
13.2 Integration 1580
13.2.1 Ethernet Switch Connectivity Attributes 1580
13.2.2 Ethernet Switch Clock and Reset Management 1580
13.2.3 Ethernet Switch Pin List 1581
13.3 Functional Description 1583
13.3.1 CPSW_3G 1583
13.3.2 Common Platform Time Sync 1607
13.3.3 CPPI Buffer Descriptors 1611
13.3.4 MDIO 1615
13.3.5 Interrupts 1618
13.3.6 Interrupt Pacing 1622
13.3.7 Reset Isolation 1623
13.3.8 CPSW_3G Network Statistics 1624
13.4 Software Operation 1625
13.4.1 Transmit Operation 1625
13.4.2 Receive Operation 1626
13.4.3 Initializing the MDIO Module 1627
13.4.4 Writing Data to a PHY Register 1627
13.4.5 Reading Data from a PHY Register 1628
13.4.6 Initialization and Configuration of CPSW 1628
13.5 CPSW_ALE Registers 1629
13.5.1 IDVER Register (offset = 0h) [reset = 290104h] 1630
13.5.2 PRESCALE Register (offset = 10h) [reset = 0h] 1631
13.5.3 UNKNOWN_VLAN Register (offset = 18h) [reset = 0h] 1632
13.5.4 TBLCTL Register (offset = 20h) [reset = 0h] 1633
13.5.5 TBLW2 Register (offset = 34h) [reset = 0h] 1634
13.5.6 TBLW1 Register (offset = 38h) [reset = 0h] 1635
13.5.7 TBLW0 Register (offset = 3Ch) [reset = 0h] 1636
13.5.8 PORTCTL0 Register (offset = 40h) [reset = 0h] 1637
13.5.9 PORTCTL1 Register (offset = 44h) [reset = 0h] 1638
13.5.10 PORTCTL2 Register (offset = 48h) [reset = 0h] 1639
13.5.11 PORTCTL3 Register (offset = 4Ch) [reset = 0h] 1640
13.5.12 CONTROL Register (offset = 8h) [reset = 0h] 1641
13.5.13 PORTCTL4 Register (offset = 50h) [reset = 0h] 1643
13.5.14 PORTCTL5 Register (offset = 54h) [reset = 0h] 1644
13.6 CPSW_CPDMA Registers 1644
13.6.1 TX_IDVER Register (offset = 0h) [reset = 180108h] 1648
13.6.2 TX_INTSTAT_RAW Register (offset = 80h) [reset = 0h] 1649

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.3 TX_INTSTAT_MASKED Register (offset = 84h) [reset = 0h] 1650
13.6.4 TX_INTMASK_SET Register (offset = 88h) [reset = 0h] 1651
13.6.5 TX_INTMASK_CLEAR Register (offset = 8Ch) [reset = 0h] 1652
13.6.6 CPDMA_IN_VECTOR Register (offset = 90h) [reset = 0h] 1653
13.6.7 CPDMA_EOI_VECTOR Register (offset = 94h) [reset = 0h] 1654
13.6.8 RX_IDVER Register (offset = 10h) [reset = C0107h] 1655
13.6.9 RX_INTSTAT_RAW Register (offset = A0h) [reset = 0h] 1656
13.6.10 RX_INTSTAT_MASKED Register (offset = A4h) [reset = 0h] 1657
13.6.11 RX_INTMASK_SET Register (offset = A8h) [reset = 0h] 1658
13.6.12 RX_INTMASK_CLEAR Register (offset = ACh) [reset = 0h] 1659
13.6.13 DMA_INTSTAT_RAW Register (offset = B0h) [reset = 0h] 1660
13.6.14 DMA_INTSTAT_MASKED Register (offset = B4h) [reset = 0h] 1661
13.6.15 DMA_INTMASK_SET Register (offset = B8h) [reset = 0h] 1662
13.6.16 DMA_INTMASK_CLEAR Register (offset = BCh) [reset = 0h] 1663
13.6.17 RX0_PENDTHRESH Register (offset = C0h) [reset = 0h] 1664
13.6.18 RX1_PENDTHRESH Register (offset = C4h) [reset = 0h] 1665
13.6.19 RX_CONTROL Register (offset = 14h) [reset = 0h] 1666
13.6.20 RX2_PENDTHRESH Register (offset = C8h) [reset = 0h] 1667
13.6.21 RX3_PENDTHRESH Register (offset = CCh) [reset = 0h] 1668
13.6.22 RX4_PENDTHRESH Register (offset = D0h) [reset = 0h] 1669
13.6.23 RX5_PENDTHRESH Register (offset = D4h) [reset = 0h] 1670
13.6.24 RX6_PENDTHRESH Register (offset = D8h) [reset = 0h] 1671
13.6.25 RX7_PENDTHRESH Register (offset = DCh) [reset = 0h] 1672
13.6.26 RX0_FREEBUFFER Register (offset = E0h) [reset = 0h] 1673
13.6.27 RX1_FREEBUFFER Register (offset = E4h) [reset = 0h] 1674
13.6.28 RX2_FREEBUFFER Register (offset = E8h) [reset = 0h] 1675
13.6.29 RX3_FREEBUFFER Register (offset = ECh) [reset = 0h] 1676
13.6.30 RX_TEARDOWN Register (offset = 18h) [reset = 0h] 1677
13.6.31 RX4_FREEBUFFER Register (offset = F0h) [reset = 0h] 1678
13.6.32 RX5_FREEBUFFER Register (offset = F4h) [reset = 0h] 1679
13.6.33 RX6_FREEBUFFER Register (offset = F8h) [reset = 0h] 1680
13.6.34 RX7_FREEBUFFER Register (offset = FCh) [reset = 0h] 1681
13.6.35 TX0_HDP Register (offset = A00h) [reset = 0h] 1682
13.6.36 TX1_HDP Register (offset = A04h) [reset = 0h] 1683
13.6.37 TX2_HDP Register (offset = A08h) [reset = 0h] 1684
13.6.38 TX3_HDP Register (offset = A0Ch) [reset = 0h] 1685
13.6.39 TX4_HDP Register (offset = A10h) [reset = 0h] 1686
13.6.40 TX5_HDP Register (offset = A14h) [reset = 0h] 1687
13.6.41 TX6_HDP Register (offset = A18h) [reset = 0h] 1688
13.6.42 TX7_HDP Register (offset = A1Ch) [reset = 0h] 1689
13.6.43 RX0_HDP Register (offset = A20h) [reset = 0h] 1690
13.6.44 RX1_HDP Register (offset = A24h) [reset = 0h] 1691
13.6.45 RX2_HDP Register (offset = A28h) [reset = 0h] 1692
13.6.46 RX3_HDP Register (offset = A2Ch) [reset = 0h] 1693
13.6.47 RX4_HDP Register (offset = A30h) [reset = 0h] 1694
13.6.48 RX5_HDP Register (offset = A34h) [reset = 0h] 1695
13.6.49 RX6_HDP Register (offset = A38h) [reset = 0h] 1696
13.6.50 RX7_HDP Register (offset = A3Ch) [reset = 0h] 1697
13.6.51 TX0_CP Register (offset = A40h) [reset = 0h] 1698
13.6.52 TX1_CP Register (offset = A44h) [reset = 0h] 1699
13.6.53 TX2_CP Register (offset = A48h) [reset = 0h] 1700
13.6.54 TX3_CP Register (offset = A4Ch) [reset = 0h] 1701
13.6.55 TX4_CP Register (offset = A50h) [reset = 0h] 1702

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.56 TX5_CP Register (offset = A54h) [reset = 0h] 1703
13.6.57 TX6_CP Register (offset = A58h) [reset = 0h] 1704
13.6.58 TX7_CP Register (offset = A5Ch) [reset = 0h] 1705
13.6.59 RX0_CP Register (offset = A60h) [reset = 0h] 1706
13.6.60 RX1_CP Register (offset = A64h) [reset = 0h] 1707
13.6.61 RX2_CP Register (offset = A68h) [reset = 0h] 1708
13.6.62 RX3_CP Register (offset = A6Ch) [reset = 0h] 1709
13.6.63 RX4_CP Register (offset = A70h) [reset = 0h] 1710
13.6.64 RX5_CP Register (offset = A74h) [reset = 0h] 1711
13.6.65 RX6_CP Register (offset = A78h) [reset = 0h] 1712
13.6.66 RX7_CP Register (offset = A7Ch) [reset = 0h] 1713
13.6.67 CPDMA_SOFT_RESET Register (offset = 1Ch) [reset = 0h] 1714
13.6.68 DMACONTROL Register (offset = 20h) [reset = 0h] 1715
13.6.69 DMASTATUS Register (offset = 24h) [reset = 0h] 1717
13.6.70 TX_CONTROL Register (offset = 4h) [reset = 0h] 1718
13.6.71 RX_BUFFER_OFFSET Register (offset = 28h) [reset = 0h] 1719
13.6.72 EMCONTROL Register (offset = 2Ch) [reset = 0h] 1720
13.6.73 TX_PRI0_RATE Register (offset = 30h) [reset = 0h] 1721
13.6.74 TX_PRI1_RATE Register (offset = 34h) [reset = 0h] 1722
13.6.75 TX_PRI2_RATE Register (offset = 38h) [reset = 0h] 1723
13.6.76 TX_PRI3_RATE Register (offset = 3Ch) [reset = 0h] 1724
13.6.77 TX_PRI4_RATE Register (offset = 40h) [reset = 0h] 1725
13.6.78 TX_PRI5_RATE Register (offset = 44h) [reset = 0h] 1726
13.6.79 TX_PRI6_RATE Register (offset = 48h) [reset = 0h] 1727
13.6.80 TX_PRI7_RATE Register (offset = 4Ch) [reset = 0h] 1728
13.6.81 TX_TEARDOWN Register (offset = 8h) [reset = 0h] 1729
13.7 CPSW_CPTS Registers 1729
13.7.1 CPTS_IDVER Register (offset = h + (0h)) [reset = 4E8A0101h] 1731
13.7.2 CPTS_TS_PUSH Register (offset = h + (Ch)) [reset = 0h] 1732
13.7.3 CPTS_TS_LOAD_VAL Register (offset = h + (10h)) [reset = 0h] 1733
13.7.4 CPTS_TS_LOAD_EN Register (offset = h + (14h)) [reset = 0h] 1734
13.7.5 CPTS_INTSTAT_RAW Register (offset = h + (20h)) [reset = 0h] 1735
13.7.6 CPTS_INTSTAT_MASKED Register (offset = h + (24h)) [reset = 0h] 1736
13.7.7 CPTS_CONTROL Register (offset = h + (4h)) [reset = 0h] 1737
13.7.8 CPTS_INT_ENABLE Register (offset = h + (28h)) [reset = 0h] 1738
13.7.9 CPTS_EVENT_POP Register (offset = h + (30h)) [reset = 0h] 1739
13.7.10 CPTS_EVENT_LOW Register (offset = h + (34h)) [reset = 0h] 1740
13.7.11 CPTS_EVENT_HIGH Register (offset = h + (38h)) [reset = 0h] 1741
13.7.12 CPTS_RFTCLK_SEL Register (offset = h + (8h)) [reset = 0h] 1742
13.8 CPSW_PORT Registers 1742
13.8.1 P0_CONTROL Register (offset = 0h) [reset = 0h] 1745
13.8.2 P0_BLK_CNT Register (offset = Ch) [reset = 41h] 1746
13.8.3 P0_TX_IN_CTL Register (offset = 10h) [reset = 40C0h] 1747
13.8.4 P0_PORT_VLAN Register (offset = 14h) [reset = 0h] 1748
13.8.5 P0_TX_PRI_MAP Register (offset = 18h) [reset = 33221001h] 1749
13.8.6 P1_CONTROL Register (offset = 100h) [reset = 0h] 1750
13.8.7 P1_MAX_BLKS Register (offset = 108h) [reset = 113h] 1752
13.8.8 P1_BLK_CNT Register (offset = 10Ch) [reset = 41h] 1753
13.8.9 P1_TX_IN_CTL Register (offset = 110h) [reset = 80040C0h] 1754
13.8.10 P1_PORT_VLAN Register (offset = 114h) [reset = 0h] 1755
13.8.11 P0_CPDMA_TX_PRI_MAP Register (offset = 1Ch) [reset = 76543210h] 1756
13.8.12 P1_TX_PRI_MAP Register (offset = 118h) [reset = 33221001h] 1757
13.8.13 P1_TS_SEQ_MTYPE Register (offset = 11Ch) [reset = 1E0000h] 1758

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.14 P1_SA_LO Register (offset = 120h) [reset = 0h] 1759
13.8.15 P1_SA_HI Register (offset = 124h) [reset = 0h] 1760
13.8.16 P1_SEND_PERCENT Register (offset = 128h) [reset = 0h] 1761
13.8.17 P1_RX_DSCP_PRI_MAP0 Register (offset = 130h) [reset = 0h] 1762
13.8.18 P1_RX_DSCP_PRI_MAP1 Register (offset = 134h) [reset = 0h] 1763
13.8.19 P1_RX_DSCP_PRI_MAP2 Register (offset = 138h) [reset = 0h] 1764
13.8.20 P1_RX_DSCP_PRI_MAP3 Register (offset = 13Ch) [reset = 0h] 1765
13.8.21 P0_CPDMA_RX_CH_MAP Register (offset = 20h) [reset = 0h] 1766
13.8.22 P1_RX_DSCP_PRI_MAP4 Register (offset = 140h) [reset = 0h] 1767
13.8.23 P1_RX_DSCP_PRI_MAP5 Register (offset = 144h) [reset = 0h] 1768
13.8.24 P1_RX_DSCP_PRI_MAP6 Register (offset = 148h) [reset = 0h] 1769
13.8.25 P1_RX_DSCP_PRI_MAP7 Register (offset = 14Ch) [reset = 0h] 1770
13.8.26 P0_RX_DSCP_PRI_MAP0 Register (offset = 30h) [reset = 0h] 1771
13.8.27 P2_CONTROL Register (offset = 200h) [reset = 0h] 1772
13.8.28 P0_RX_DSCP_PRI_MAP1 Register (offset = 34h) [reset = 0h] 1774
13.8.29 P2_MAX_BLKS Register (offset = 208h) [reset = 113h] 1775
13.8.30 P2_BLK_CNT Register (offset = 20Ch) [reset = 41h] 1776
13.8.31 P2_TX_IN_CTL Register (offset = 210h) [reset = 80040C0h] 1777
13.8.32 P2_PORT_VLAN Register (offset = 214h) [reset = 0h] 1778
13.8.33 P2_TX_PRI_MAP Register (offset = 218h) [reset = 33221001h] 1779
13.8.34 P2_TS_SEQ_MTYPE Register (offset = 21Ch) [reset = 1E0000h] 1780
13.8.35 P2_SA_LO Register (offset = 220h) [reset = 0h] 1781
13.8.36 P2_SA_HI Register (offset = 224h) [reset = 0h] 1782
13.8.37 P2_SEND_PERCENT Register (offset = 228h) [reset = 0h] 1783
13.8.38 P0_RX_DSCP_PRI_MAP2 Register (offset = 38h) [reset = 0h] 1784
13.8.39 P2_RX_DSCP_PRI_MAP0 Register (offset = 230h) [reset = 0h] 1785
13.8.40 P2_RX_DSCP_PRI_MAP1 Register (offset = 234h) [reset = 0h] 1786
13.8.41 P2_RX_DSCP_PRI_MAP2 Register (offset = 238h) [reset = 0h] 1787
13.8.42 P2_RX_DSCP_PRI_MAP3 Register (offset = 23Ch) [reset = 0h] 1788
13.8.43 P2_RX_DSCP_PRI_MAP4 Register (offset = 240h) [reset = 0h] 1789
13.8.44 P2_RX_DSCP_PRI_MAP5 Register (offset = 244h) [reset = 0h] 1790
13.8.45 P2_RX_DSCP_PRI_MAP6 Register (offset = 248h) [reset = 0h] 1791
13.8.46 P2_RX_DSCP_PRI_MAP7 Register (offset = 24Ch) [reset = 0h] 1792
13.8.47 P0_RX_DSCP_PRI_MAP3 Register (offset = 3Ch) [reset = 0h] 1793
13.8.48 P0_RX_DSCP_PRI_MAP4 Register (offset = 40h) [reset = 0h] 1794
13.8.49 P0_RX_DSCP_PRI_MAP5 Register (offset = 44h) [reset = 0h] 1795
13.8.50 P0_RX_DSCP_PRI_MAP6 Register (offset = 48h) [reset = 0h] 1796
13.8.51 P0_RX_DSCP_PRI_MAP7 Register (offset = 4Ch) [reset = 0h] 1797
13.8.52 P0_MAX_BLKS Register (offset = 8h) [reset = 104h] 1798
13.9 CPSW_SL Registers 1798
13.9.1 IDVER Register (offset = 0h) [reset = 170112h] 1800
13.9.2 SOFT_RESET Register (offset = Ch) [reset = 0h] 1801
13.9.3 RX_MAXLEN Register (offset = 10h) [reset = 5EEh] 1802
13.9.4 BOFFTEST Register (offset = 14h) [reset = 0h] 1803
13.9.5 RX_PAUSE Register (offset = 18h) [reset = 0h] 1804
13.9.6 TX_PAUSE Register (offset = 1Ch) [reset = 0h] 1805
13.9.7 EMCONTROL Register (offset = 20h) [reset = 0h] 1806
13.9.8 RX_PRI_MAP Register (offset = 24h) [reset = 76543210h] 1807
13.9.9 MACCONTROL Register (offset = 4h) [reset = 0h] 1808
13.9.10 TX_GAP Register (offset = 28h) [reset = Ch] 1810
13.9.11 MACSTATUS Register (offset = 8h) [reset = 80000000h] 1811
13.10 CPSW_SS Registers 1811
13.10.1 ID_VER Register (offset = 0h) [reset = 190112h] 1813

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.10.2 STAT_PORT_EN Register (offset = Ch) [reset = 0h] 1814
13.10.3 PTYPE Register (offset = 10h) [reset = 0h] 1815
13.10.4 SOFT_IDLE Register (offset = 14h) [reset = 0h] 1816
13.10.5 THRU_RATE Register (offset = 18h) [reset = 3003h] 1817
13.10.6 GAP_THRESH Register (offset = 1Ch) [reset = Bh] 1818
13.10.7 TX_START_WDS Register (offset = 20h) [reset = 20h] 1819
13.10.8 FLOW_CONTROL Register (offset = 24h) [reset = 1h] 1820
13.10.9 CONTROL Register (offset = 4h) [reset = 0h] 1821
13.10.10 VLAN_LTYPE Register (offset = 28h) [reset = 81008100h] 1822
13.10.11 TS_LTYPE Register (offset = 2Ch) [reset = 0h] 1823
13.10.12 DLR_LTYPE Register (offset = 30h) [reset = 80E1h] 1824
13.10.13 SOFT_RESET Register (offset = 8h) [reset = 0h] 1825
13.11 CPSW_WR Registers 1825
13.11.1 IDVER Register (offset = 0h) [reset = EEB0100h] 1827
13.11.2 C2_RX_STAT Register (offset = 64h) [reset = 0h] 1828
13.11.3 C2_TX_STAT Register (offset = 68h) [reset = 0h] 1829
13.11.4 C2_MISC_STAT Register (offset = 6Ch) [reset = 0h] 1830
13.11.5 C0_RX_IMAX Register (offset = 70h) [reset = 0h] 1831
13.11.6 C0_TX_IMAX Register (offset = 74h) [reset = 0h] 1832
13.11.7 INT_CONTROL Register (offset = Ch) [reset = 0h] 1833
13.11.8 C1_RX_IMAX Register (offset = 78h) [reset = 0h] 1834
13.11.9 C1_TX_IMAX Register (offset = 7Ch) [reset = 0h] 1835
13.11.10 C2_RX_IMAX Register (offset = 80h) [reset = 0h] 1836
13.11.11 C2_TX_IMAX Register (offset = 84h) [reset = 0h] 1837
13.11.12 RGMII_CTL Register (offset = 88h) [reset = 0h] 1838
13.11.13 C0_RX_THRESH_EN Register (offset = 10h) [reset = 0h] 1839
13.11.14 C0_RX_EN Register (offset = 14h) [reset = 0h] 1840
13.11.15 C0_TX_EN Register (offset = 18h) [reset = 0h] 1841
13.11.16 C0_MISC_EN Register (offset = 1Ch) [reset = 0h] 1842
13.11.17 C1_RX_THRESH_EN Register (offset = 20h) [reset = 0h] 1843
13.11.18 C1_RX_EN Register (offset = 24h) [reset = 0h] 1844
13.11.19 SOFT_RESET Register (offset = 4h) [reset = 0h] 1845
13.11.20 C1_TX_EN Register (offset = 28h) [reset = 0h] 1846
13.11.21 C1_MISC_EN Register (offset = 2Ch) [reset = 0h] 1847
13.11.22 C2_RX_THRESH_EN Register (offset = 30h) [reset = 0h] 1848
13.11.23 C2_RX_EN Register (offset = 34h) [reset = 0h] 1849
13.11.24 C2_TX_EN Register (offset = 38h) [reset = 0h] 1850
13.11.25 C2_MISC_EN Register (offset = 3Ch) [reset = 0h] 1851
13.11.26 C0_RX_THRESH_STAT Register (offset = 40h) [reset = 0h] 1852
13.11.27 C0_RX_STAT Register (offset = 44h) [reset = 0h] 1853
13.11.28 C0_TX_STAT Register (offset = 48h) [reset = 0h] 1854
13.11.29 C0_MISC_STAT Register (offset = 4Ch) [reset = 0h] 1855
13.11.30 CONTROL Register (offset = 8h) [reset = Ah] 1856
13.11.31 C1_RX_THRESH_STAT Register (offset = 50h) [reset = 0h] 1857
13.11.32 C1_RX_STAT Register (offset = 54h) [reset = 0h] 1858
13.11.33 C1_TX_STAT Register (offset = 58h) [reset = 0h] 1859
13.11.34 C1_MISC_STAT Register (offset = 5Ch) [reset = 0h] 1860
13.11.35 C2_RX_THRESH_STAT Register (offset = 60h) [reset = 0h] 1861
14 Pulse-Width Modulation Subsystem (PWMSS) 1863
14.1 Pulse-Width Modulation Subsystem (PWMSS) 1864
14.1.1 Introduction 1864
14.1.2 Integration 1866
14.2 Enhanced High-Resolution PWM (eHRPWM) Module 1868

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.1 Introduction 1868
14.2.2 Functional Description 1874
14.2.3 Use Cases 1944
14.2.4 Registers 1968
14.3 Enhanced Capture (eCAP) Module 1969
14.3.1 Introduction 1969
14.3.2 Functional Description 1970
14.3.3 Use Cases 1980
14.3.4 eCAP Module Control and Status Registers 1996
14.4 Enhanced Quadrature Encoder Pulse (eQEP) Module 2008
14.4.1 Introduction 2008
14.4.2 Functional Description 2011
14.4.3 eQEP Registers 2029

15 Universal Serial Bus (USB) 2047
15.1 Introduction 2048
15.1.1 Acronyms, Abbreviations, and Definitions 2048
15.1.2 Unsupported USB OTG and PHY Features 2050
15.2 Integration 2051
15.2.1 USB Connectivity Attributes 2051
15.2.2 USB Clock and Reset Management 2051
15.2.3 USB Pin List 2051
15.2.4 USB GPIO Details 2052
15.2.5 USB Unbonded PHY Pads 2052
15.3 Functional Description 2053
15.3.1 VBUS Voltage Sourcing Control 2053
15.3.2 Pull-up/Pull-Down Resistors 2053
15.3.3 Role Assuming Method 2054
15.3.4 Clock, PLL, and PHY Initialization 2054
15.3.5 Indexed and Non-Indexed Register Spaces 2055
15.3.6 Dynamic FIFO Sizing 2055
15.3.7 USB Controller Host and Peripheral Modes Operation 2056
15.3.8 Protocol Description(s) 2057
15.3.9 Communications Port Programming Interface (CPPI) 4.1 DMA 2090
15.3.10 USB 2.0 Test Modes 2114
15.4 Supported Use Cases 2115
15.5 USB Registers 2116
15.5.1 USBSS Registers 2116
15.5.2 USB0 and USB1 Controller Registers 2148
15.5.3 CPPI DMA Controller Registers 2212
15.5.4 CPPI DMA Scheduler Registers 2219
15.5.5 CPPI DMA Queue Manager Registers 2222
15.5.6 USB Mentor Core Registers 2242
15.5.7 FIFOs 2268

16 Interprocessor Communication 2279
16.1 Mailbox 2280
16.1.1 Introduction 2280
16.1.2 Integration 2281
16.2 Spinlock 2282
16.2.1 Introduction 2282
16.2.2 SPINLOCK Registers 2282

17 Multimedia Card (MMC) 2321
17.1 Introduction 2322
17.1.1 MMCHS Features 2322

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.1.2 Unsupported MMCHS Features 2322
17.2 Integration 2323
17.2.1 MMCHS Connectivity Attributes 2323
17.2.2 MMCHS Clock and Reset Management 2323
17.2.3 MMCHS Pin List 2324
17.3 Functional Description 2326
17.3.1 MMC/SD/SDIO Functional Modes 2326
17.3.2 Resets 2333
17.3.3 Power Management 2334
17.3.4 Interrupt Requests 2337
17.3.5 DMA Modes 2339
17.3.6 Mode Selection 2342
17.3.7 Buffer Management 2342
17.3.8 Transfer Process 2345
17.3.9 Transfer or Command Status and Error Reporting 2346
17.3.10 Auto Command 12 Timings 2351
17.3.11 Transfer Stop 2353
17.3.12 Output Signals Generation 2354
17.3.13 Card Boot Mode Management 2356
17.3.14 CE-ATA Command Completion Disable Management 2357
17.3.15 Test Registers 2357
17.3.16 MMC/SD/SDIO Hardware Status Features 2358
17.3.17 Low-Level Programming Models 2359
17.4 Use Cases 2363
17.5 MMC/SD/SDIO Registers 2364
17.5.1 IP Revision Identifier Register (SD_HL_REV) 2365
17.5.2 IP Module Hardware Configuration Register (SD_HL_HWINFO) 2365
17.5.3 Clock Management Configuration (SD_HL_SYSCONFIG) 2365
17.5.4 System Configuration Register (SD_SYSCONFIG) 2366
17.5.5 System Status Register (SD_SYSSTATUS) 2368
17.5.6 Card Status Response Error (SD_CSRE) 2368
17.5.7 System Test Register (SD_SYSTEST) 2369
17.5.8 Configuration Register (SD_CON) 2372
17.5.9 Power Counter Register (SD_PWCNT) 2375
17.5.10 SDMA System Address (SD_SDMASA) 2375
17.5.11 Transfer Length Configuration Register (SD_BLK) 2376
17.5.12 Command Argument Register (SD_ARG) 2377
17.5.13 Command and Transfer Mode Register (SD_CMD) 2377
17.5.14 Command Response[31:0] Register (SD_RSP10) 2381
17.5.15 Command Response[63:32] Register (SD_RSP32) 2381
17.5.16 Command Response[95:64] Register (SD_RSP54) 2382
17.5.17 Command Response[127:96] Register (SD_RSP76) 2382
17.5.18 Data Register (SD_DATA) 2383
17.5.19 Present State Register (SD_PSTATE) 2384
17.5.20 Control Register (SD_HCTL) 2387
17.5.21 SD System Control Register (SD_SYSCTL) 2390
17.5.22 Interrupt Status Register (SD_STAT) 2392
17.5.23 Interrupt SD Enable Register (SD_IE) 2397
17.5.24 Interrupt Signal Enable Register (SD_ISE) 2400
17.5.25 Auto CMD12 Error Status Register (SD_AC12) 2403
17.5.26 Capabilities Register (SD_CAPA) 2404
17.5.27 Maximum Current Capabilities Register (SD_CUR_CAPA) 2406
17.5.28 Force Event Register for Error Interrupt Status (SD_FE) 2407

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.5.29 ADMA Error Status Register (SD_ADMAES) 2409
17.5.30 ADMA System Address Low Bits Register (SD_ADMASAL) 2410
17.5.31 ADMA System Address High Bits Register (SD_ADMASAH) 2410
17.5.32 Versions Register (SD_REV) 2411

18 Universal Asynchronous Receiver/Transmitter (UART) 2413

18.1 Introduction 2414
18.1.1 UART Mode Features 2414
18.1.2 IrDA Mode Features 2414
18.1.3 Unsupported UART Features 2414
18.2 Intregration 2416
18.2.1 UART Connectivity Attributes 2416
18.2.2 UART Clock and Reset Management 2416
18.2.3 UART Pin List 2418
18.3 Functional Description 2419
18.3.1 Block Diagram 2419
18.3.2 Clock Configuration 2420
18.3.3 Software Reset 2420
18.3.4 Power Management 2420
18.3.5 Interrupt Requests 2422
18.3.6 FIFO Management 2425
18.3.7 Mode Selection 2434
18.3.8 Protocol Formatting 2440
18.4 UART/IrDA/CIR Basic Programming Model 2453
18.4.1 UART Programming Model 2453
18.4.2 IrDA Programming Model (UART3 Only) 2459
18.5 UART Registers 2462
18.5.1 Receiver Holding Register (RHR) 2463
18.5.2 Transmit Holding Register (THR) 2463
18.5.3 Interrupt Enable Register (IER) - UART Mode 2464
18.5.4 Interrupt Enable Register (IER) - IrDA Mode 2465
18.5.5 Interrupt Enable Register (IER) - CIR Mode 2466
18.5.6 Interrupt Identification Register (IIR) - UART Mode 2467
18.5.7 Interrupt Identification Register (IIR) - IrDA Mode 2468
18.5.8 Interrupt Identification Register (IIR) - CIR Mode 2469
18.5.9 FIFO Control Register (FCR) 2470
18.5.10 Line Control Register (LCR) 2471
18.5.11 Modem Control Register (MCR) 2472
18.5.12 Line Status Register (LSR) - UART Mode 2473
18.5.13 Line Status Register (LSR) - IrDA Mode 2474
18.5.14 Line Status Register (LSR) - CIR Mode 2475
18.5.15 Modem Status Register (MSR) 2476
18.5.16 Transmission Control Register (TCR) 2477
18.5.17 Scratchpad Register (SPR) 2477
18.5.18 Trigger Level Register (TLR) 2478
18.5.19 Mode Definition Register 1 (MDR1) 2479
18.5.20 Mode Definition Register 2 (MDR2) 2480
18.5.21 Status FIFO Line Status Register (SFLSR) 2481
18.5.22 RESUME Register 2481
18.5.23 Status FIFO Register Low (SFREGL) 2482
18.5.24 Status FIFO Register High (SFREGH) 2482
18.5.25 BOF Control Register (BLR) 2483
18.5.26 Auxiliary Control Register (ACREG) 2484
18.5.27 Supplementary Control Register (SCR) 2485

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.28 Supplementary Status Register (SSR) 2486
18.5.29 BOF Length Register (EBLR) 2487
18.5.30 Module Version Register (MVR) 2488
18.5.31 System Configuration Register (SYSC) 2489
18.5.32 System Status Register (SYSS) 2489
18.5.33 Wake-Up Enable Register (WER) 2490
18.5.34 Carrier Frequency Prescaler Register (CFPS) 2491
18.5.35 Divisor Latches Low Register (DLL) 2492
18.5.36 Divisor Latches High Register (DLH) 2492
18.5.37 Enhanced Feature Register (EFR) 2493
18.5.38 XON1/ADDR1 Register 2494
18.5.39 XON2/ADDR2 Register 2494
18.5.40 XOFF1 Register 2495
18.5.41 XOFF2 Register 2495
18.5.42 Transmit Frame Length Low Register (TXFLL) 2496
18.5.43 Transmit Frame Length High Register (TXFLH) 2496
18.5.44 Received Frame Length Low Register (RXFLL) 2497
18.5.45 Received Frame Length High Register (RXFLH) 2497
18.5.46 UART Autobauding Status Register (UASR) 2498

19 Timer 2499
19.1 DMTimer 2500
19.1.1 Introduction 2500
19.1.2 Integration 2502
19.1.3 Functional Description 2504
19.1.4 Use Cases 2512
19.1.5 Timer Registers 2513
19.2 DMTimer 1ms 2527
19.2.1 Introduction 2527
19.2.2 Integration 2528
19.2.3 Functional Description 2529
19.2.4 Use Cases 2534
19.2.5 DMTIMER_1MS Registers 2534
19.3 RTC_SS 2556
19.3.1 Introduction 2556
19.3.2 Integration 2557
19.3.3 Functional Description 2559
19.3.4 Use Cases 2566
19.3.5 RTC Registers 2567
19.4 WATCHDOG 2585
19.4.1 Introduction 2585
19.4.2 Integration 2586
19.4.3 Functional Description 2587
19.4.4 Use Cases 2594
19.4.5 Watchdog Timer Registers 2595

20 I2C 2609
20.1 Introduction 2610
20.1.1 I2C Features 2610
20.1.2 Unsupported I2C Features 2610
20.2 Integration 2611
20.2.1 I2C Connectivity Attributes 2611
20.2.2 I2C Clock and Reset Management 2611
20.2.3 I2C Pin List 2612
20.3 Functional Description 2613

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.3.1 Functional Block Diagram 2613
20.3.2 I2C Master/Slave Contoller Signals 2613
20.3.3 I2C Reset 2614
20.3.4 Data Validity 2614
20.3.5 START & STOP Conditions 2616
20.3.6 I2C Operation 2616
20.3.7 Arbitration 2618
20.3.8 I2C Clock Generation and I2C Clock Synchronization 2618
20.3.9 Prescaler (SCLK/ICLK) 2619
20.3.10 Noise Filter 2619
20.3.11 I2C Interrupts 2619
20.3.12 DMA Events 2620
20.3.13 Interrupt and DMA Events 2620
20.3.14 FIFO Management 2620
20.3.15 How to Program I2C 2624
20.4 I2C Registers 2626
20.4.1 Module Revision Register (LOW BYTES) (I2C_REVNB_LO) 2626
20.4.2 Module Revision Register (HIGH BYTES) (I2C_REVNB_HI) 2628
20.4.3 System Configuration Register (I2C_SYSC) 2629
20.4.4 I2C End of Interrupt Register (I2C_EOI) 2630
20.4.5 I2C Status Raw Register (I2C_IRQSTATUS_RAW) 2630
20.4.6 I2C Status Register (I2C_IRQSTATUS) 2635
20.4.7 I2C Interrupt Enable Set Register (I2C_IRQENABLE_SET) 2637
20.4.8 I2C Interrupt Enable Clear Register (I2C_IRQENABLE_CLR) 2639
20.4.9 I2C Wakeup Enable Register (I2C_WE) 2641
20.4.10 Receive DMA Enable Set Register (I2C_DMARXENABLE_SET) 2644
20.4.11 Transmit DMA Enable Set Register (I2C_DMATXENABLE_SET) 2644
20.4.12 Receive DMA Enable Clear Register (I2C_DMARXENABLE_CLR) 2645
20.4.13 Transmit DMA Enable Clear Register (I2C_DMATXENABLE_CLR) 2645
20.4.14 Receive DMA Wakeup Register (I2C_DMARXWAKE_EN) 2646
20.4.15 Transmit DMA Wakeup Register (I2C_DMATXWAKE_EN) 2648
20.4.16 System Status Register (I2C_SYSS) 2650
20.4.17 Buffer Configuration Register (I2C_BUF) 2650
20.4.18 Data Counter Register (I2C_CNT) 2652
20.4.19 Data Access Register (I2C_DATA) 2653
20.4.20 I2C Configuration Register (I2C_CON) 2654
20.4.21 I2C Own Address Register (I2C_OA) 2656
20.4.22 I2C Slave Address Register (I2C_SA) 2657
20.4.23 I2C Clock Prescaler Register (I2C_PSC) 2658
20.4.24 I2C SCL Low Time Register (I2C_SCLL) 2659
20.4.25 I2C SCL High Time Register (I2C_SCLH) 2659
20.4.26 System Test Register (I2C_SYSTEST) 2660
20.4.27 I2C Buffer Status Register (I2C_BUFSTAT) 2663
20.4.28 Own Address 1 (OA1) (I2C_OA1) 2664
20.4.29 I2C Own Address 2 Register (I2C_OA2) 2665
20.4.30 I2C Own Address 3 Register (I2C_OA3) 2666
20.4.31 Active Own Address Register (I2C_ACTOA) 2667
20.4.32 I2C Clock Blocking Enable Register (I2C_SBLOCK) 2668

21 Multichannel Audio Serial Port (McASP) 2669
21.1 Introduction 2670
21.1.1 Purpose of the Peripheral 2670
21.1.2 Features 2670
21.1.3 Protocols Supported 2670

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.1.4 Unsupported McASP Features 2671
21.2 Integration 2672
21.2.1 McASP Connectivity Attributes 2672
21.2.2 McASP Clock and Reset Management 2673
21.2.3 McASP Pin List 2673
21.3 Functional Description 2674
21.3.1 Overview 2674
21.3.2 Clock and Frame Sync Generators 2674
21.3.3 Memory Map 2678
21.3.4 Signal Descriptions 2678
21.3.5 Pin Multiplexing 2678
21.3.6 Transfer Modes 2679
21.3.7 General Architecture 2686
21.3.8 Operation 2690
21.3.9 Reset Considerations 2707
21.3.10 Setup and Initialization 2707
21.3.11 Interrupts 2712
21.3.12 EDMA Event Support 2714
21.3.13 Power Management 2716
21.3.14 Emulation Considerations 2716
21.4 McASP Registers 2717
21.4.1 Revision Identification Register (REV) 2720
21.4.2 Pin Function Register (PFUNC) 2721
21.4.3 Pin Direction Register (PDIR) 2723
21.4.4 Pin Data Output Register (PDOUT) 2725
21.4.5 Pin Data Input Register (PDIN) 2727
21.4.6 Pin Data Set Register (PDSET) 2729
21.4.7 Pin Data Clear Register (PDCLR) 2731
21.4.8 Global Control Register (GBLCTL) 2733
21.4.9 Audio Mute Control Register (AMUTE) 2735
21.4.10 Digital Loopback Control Register (DLBCTL) 2737
21.4.11 Digital Mode Control Register (DITCTL) 2738
21.4.12 Receiver Global Control Register (RGBLCTL) 2739
21.4.13 Receive Format Unit Bit Mask Register (RMASK) 2740
21.4.14 Receive Bit Stream Format Register (RFMT) 2741
21.4.15 Receive Frame Sync Control Register (AFSRCTL) 2743
21.4.16 Receive Clock Control Register (ACLKRCTL) 2744
21.4.17 Receive High-Frequency Clock Control Register (AHCLKRCTL) 2745
21.4.18 Receive TDM Time Slot Register (RTDM) 2746
21.4.19 Receiver Interrupt Control Register (RINTCTL) 2747
21.4.20 Receiver Status Register (RSTAT) 2748
21.4.21 Current Receive TDM Time Slot Registers (RSLOT) 2749
21.4.22 Receive Clock Check Control Register (RCLKCHK) 2750
21.4.23 Receiver DMA Event Control Register (REVTCTL) 2751
21.4.24 Transmitter Global Control Register (XGBLCTL) 2752
21.4.25 Transmit Format Unit Bit Mask Register (XMASK) 2753
21.4.26 Transmit Bit Stream Format Register (XFMT) 2754
21.4.27 Transmit Frame Sync Control Register (AFSXCTL) 2756
21.4.28 Transmit Clock Control Register (ACLKXCTL) 2757
21.4.29 Transmit High-Frequency Clock Control Register (AHCLKXCTL) 2758
21.4.30 Transmit TDM Time Slot Register (XTDM) 2759
21.4.31 Transmitter Interrupt Control Register (XINTCTL) 2760
21.4.32 Transmitter Status Register (XSTAT) 2761

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.33 Current Transmit TDM Time Slot Register (XSLOT) 2762
21.4.34 Transmit Clock Check Control Register (XCLKCHK) 2763
21.4.35 Transmitter DMA Event Control Register (XEVTCTL) 2764
21.4.36 Serializer Control Registers (SRCTL*n*) 2765
21.4.37 DIT Left Channel Status Registers (DITCSRA0-DITCSRA5) 2766
21.4.38 DIT Right Channel Status Registers (DITCSRB0-DITCSRB5) 2766
21.4.39 DIT Left Channel User Data Registers (DITUDRA0-DITUDRA5) 2766
21.4.40 DIT Right Channel User Data Registers (DITUDRB0-DITUDRB5) 2766
21.4.41 Transmit Buffer Registers (XBUF*n*) 2767
21.4.42 Receive Buffer Registers (RBUF*n*) 2767
21.4.43 Write FIFO Control Register (WFIFOCTL) 2768
21.4.44 Write FIFO Status Register (WFIFOSTS) 2769
21.4.45 Read FIFO Control Register (RFIFOCTL) 2770
21.4.46 Read FIFO Status Register (RFIFOSTS) 2771

22 Controller Area Network (CAN) 2773

22.1 Introduction 2774
22.1.1 DCAN Features 2774
22.1.2 Unsupported DCAN Features 2774
22.2 Integration 2775
22.2.1 DCAN Connectivity Attributes 2775
22.2.2 DCAN Clock and Reset Management 2775
22.2.3 DCAN Pin List 2775
22.3 Functional Description 2776
22.3.1 CAN Operation 2777
22.3.2 Dual Clock Source 2784
22.3.3 Interrupt Functionality 2785
22.3.4 Local Power-Down Mode 2787
22.3.5 Parity Check Mechanism 2789
22.3.6 Debug/Suspend Mode 2790
22.3.7 Configuration of Message Objects 2790
22.3.8 Message Handling 2793
22.3.9 CAN Bit Timing 2798
22.3.10 Message Interface Register Sets 2806
22.3.11 Message RAM 2808
22.3.12 GIO Support 2813
22.4 Use Cases 2813
22.5 DCAN Control Registers 2814
22.5.1 CAN Control Register (DCAN CTL) 2815
22.5.2 Error and Status Register (DCAN ES) and DCAN Parity Error EOI Register (PARITYERR_EOI) 2817
22.5.3 Error Counter Register (DCAN ERRC) 2819
22.5.4 Bit Timing Register (DCAN BTR) 2820
22.5.5 Interrupt Register (DCAN INT) 2821
22.5.6 Test Register (DCAN TEST) 2822
22.5.7 Parity Error Code Register (DCAN PERR) 2823
22.5.8 Auto-Bus-On Time Register (DCAN ABOTR) 2824
22.5.9 Transmission Request X Register (DCAN TXRQ X) 2825
22.5.10 Transmission Request Registers (DCAN TXRQ12 to DCAN TXRQ78) 2826
22.5.11 New Data X Register (DCAN NWDAT X) 2827
22.5.12 New Data Registers (DCAN NWDAT12 to DCAN NWDAT78) 2828
22.5.13 Interrupt Pending X Register (DCAN INTPND X) 2829
22.5.14 Interrupt Pending Registers (DCAN INTPND12 to DCAN INTPND78) 2830
22.5.15 Message Valid X Register (DCAN MSGVAL X) 2831
22.5.16 Message Valid Registers (DCAN MSGVAL12 to DCAN MSGVAL78) 2832

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.5.17 Interrupt Multiplexer Registers (DCAN INTMUX12 to DCAN INTMUX78) 2833
22.5.18 IF1/IF2 Command Registers (DCAN IF1CMD, DCAN IF2CMD) 2834
22.5.19 IF1/IF2 Mask Registers (DCAN IF1MSK, DCAN IF2MSK) 2837
22.5.20 IF1/IF2 Arbitration Registers (DCAN IF1ARB, DCAN IF2ARB) 2838
22.5.21 IF1/IF2 Message Control Registers (DCAN IF1MCTL, DCAN IF2MCTL) 2840
22.5.22 IF1/IF2 Data A and Data B Registers (DCAN IF1DATA/DATB, DCAN IF2DATA/DATB) 2842
22.5.23 IF3 Observation Register (DCAN IF3OBS) 2843
22.5.24 IF3 Mask Register (DCAN IF3MSK) 2845
22.5.25 IF3 Arbitration Register (DCAN IF3ARB) 2846
22.5.26 IF3 Message Control Register (DCAN IF3MCTL) 2847
22.5.27 IF3 Data A and Data B Registers (DCAN IF3DATA/DATB) 2849
22.5.28 Update Enable Registers (DCAN IF3UPD12 to IF3UPD78) 2850
22.5.29 CAN TX I/O Control Register (DCAN TIOC) 2851
22.5.30 CAN RX IO Control Register (DCAN RIOC) 2853

23 Multichannel Serial Port Interface (McSPI) 2855
23.1 Introduction 2856
23.1.1 McSPI Features 2856
23.1.2 Unsupported McSPI Features 2856
23.2 Integration 2856
23.2.1 McSPI Connectivity Attributes 2858
23.2.2 McSPI Clock and Reset Management 2858
23.2.3 McSPI Pin List 2858
23.3 Functional Description 2859
23.3.1 SPI Interface 2859
23.3.2 SPI Transmission 2859
23.3.3 Master Mode 2866
23.3.4 Slave Mode 2884
23.3.5 Interrupts 2888
23.3.6 DMA Requests 2889
23.3.7 Emulation Mode 2890
23.3.8 Power Saving Management 2891
23.3.9 System Test Mode 2892
23.3.10 Reset 2892
23.3.11 Access to Data Registers 2893
23.3.12 Programming Aid 2893
23.3.13 Interrupt and DMA Events 2894
23.4 Use Cases 2894
23.5 SPI Registers 2895
23.5.1 McSPI System Configuration Register (MCSPI_SYSCONFIG) 2896
23.5.2 McSPI System Status Register (MCSPI_SYSSTATUS) 2897
23.5.3 McSPI Interrupt Status Register (MCSPI_IRQSTATUS) 2898
23.5.4 McSPI Interrupt Enable Register (MCSPI_IRQENABLE) 2901
23.5.5 McSPI System Register (MCSPI_SYST) 2903
23.5.6 McSPI Module Control Register (MCSPI_MODULCTRL) 2905
23.5.7 McSPI Channel *(i)* Configuration Register (MCSPI_CH(i)CONF) 2907
23.5.8 McSPI Channel *(i)* Status Register (MCSPI_CH(i)STAT) 2911
23.5.9 McSPI Channel *(i)* Control Register (MCSPI_CH(I)CTRL) 2912
23.5.10 McSPI Channel *(i)* Transmit Register (MCSPI_TX*(i)*) 2913
23.5.11 McSPI Channel *(i)* Receive Register (MCSPI_RX*(i)*) 2913
23.5.12 McSPI Transfer Levels Register (MCSPI_XFERLEVEL) 2914

24 General-Purpose Input/Output 2915
24.1 Introduction 2916
24.1.1 Purpose of the Peripheral 2916

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

24.1.2 GPIO Features 2916
24.1.3 Unsupported GPIO Features 2916
24.2 Integration 2917
24.2.1 GPIO Connectivity Attributes 2918
24.2.2 GPIO Clock and Reset Management 2918
24.2.3 GPIO Pin List 2919
24.3 Functional Description 2920
24.3.1 Operating Modes 2920
24.3.2 Clocking and Reset Strategy 2920
24.3.3 Interrupt and Wake-up Features 2921
24.3.4 General-Purpose Interface Basic Programming Model 2925
24.4 GPIO Registers 2930
24.4.1 GPIO_REVISION Register 2931
24.4.2 GPIO_SYSCONFIG Register 2932
24.4.3 GPIO_EOI Register 2933
24.4.4 GPIO_IRQSTATUS_RAW_n Register 2934
24.4.5 GPIO_IRQSTATUS_n Register 2934
24.4.6 GPIO_IRQSTATUS_SET_n Register 2935
24.4.7 GPIO_IRQSTATUS_CLR_n Register 2935
24.4.8 GPIO_IRQWAKEN_n Register 2936
24.4.9 GPIO_SYSSTATUS Register 2937
24.4.10 GPIO_CTRL Register 2938
24.4.11 GPIO_OE Register 2938
24.4.12 GPIO_DATAIN Register 2939
24.4.13 GPIO_DATAOUT Register 2939
24.4.14 GPIO_LEVELDETECT0 Register 2940
24.4.15 GPIO_LEVELDETECT1 Register 2940
24.4.16 GPIO_RISINGDETECT Register 2941
24.4.17 GPIO_FALLINGDETECT Register 2941
24.4.18 GPIO_DEBOUNCENABLE Register 2942
24.4.19 GPIO_DEBOUNCINGTIME Register 2942
24.4.20 GPIO_CLEARDATAOUT Register 2943
24.4.21 GPIO_SETDATAOUT Register 2943

25 Initialization 2945
25.1 Introduction 2946
25.2 Integration 2946
25.3 Functional Description 2946
25.3.1 Device Types 2946
25.3.2 Architecture 2946
25.3.3 Functionality 2946
25.3.4 Memory Map 2947
25.3.5 Start-up and Configuration 2950
25.3.6 Booting 2951
25.3.7 Fast External Booting 2953
25.3.8 Memory Booting 2953
25.3.9 Peripheral Booting 2970
25.3.10 Image Format 2975
25.3.11 Authentication and Execution 2977
25.3.12 Wakeup 2978
25.3.13 Tracing 2979
25.4 Use Cases 2982
25.5 Registers 2982

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

List of Figures

2-1. INTCPS_REVISION Register 151
2-2. INTCPS_FIQ_PRIORITY Register 152
2-3. INTCPS_THRESHOLD Register 153
2-4. INTCPS_ITR0 Register 154
2-5. INTCPS_MIR0 Register 155
2-6. INTCPS_MIR_CLEAR0 Register 156
2-7. INTCPS_MIR_SET0 Register 157
2-8. INTCPS_ISR_SET0 Register 158
2-9. INTCPS_ISR_CLEAR0 Register 159
2-10. INTCPS_PENDING_IRQ0 Register 160
2-11. INTCPS_PENDING_FIQ0 Register 161
2-12. INTCPS_SYSCONFIG Register 162
2-13. INTCPS_ITR1 Register 163
2-14. INTCPS_MIR1 Register 164
2-15. INTCPS_MIR_CLEAR1 Register 165
2-16. INTCPS_MIR_SET1 Register 166
2-17. INTCPS_ISR_SET1 Register 167
2-18. INTCPS_ISR_CLEAR1 Register 168
2-19. INTCPS_PENDING_IRQ1 Register 169
2-20. INTCPS_PENDING_FIQ1 Register 170
2-21. INTCPS_ITR2 Register 171
2-22. INTCPS_MIR2 Register 172
2-23. INTCPS_SYSSTATUS Register 173
2-24. INTCPS_MIR_CLEAR2 Register 174
2-25. INTCPS_MIR_SET2 Register 175
2-26. INTCPS_ISR_SET2 Register 176
2-27. INTCPS_ISR_CLEAR2 Register 177
2-28. INTCPS_PENDING_IRQ2 Register 178
2-29. INTCPS_PENDING_FIQ2 Register 179
2-30. INTCPS_ITR3 Register 180
2-31. INTCPS_MIR3 Register 181
2-32. INTCPS_MIR_CLEAR3 Register 182
2-33. INTCPS_MIR_SET3 Register 183
2-34. INTCPS_ISR_SET3 Register 184
2-35. INTCPS_ISR_CLEAR3 Register 185
2-36. INTCPS_PENDING_IRQ3 Register 186
2-37. INTCPS_PENDING_FIQ3 Register 187
2-38. INTCPS_ILR0 Register 188
2-39. INTCPS_ILR1 Register 189
2-40. INTCPS_ILR2 Register 190
2-41. INTCPS_ILR3 Register 191
2-42. INTCPS_ILR4 Register 192
2-43. INTCPS_ILR5 Register 193
2-44. INTCPS_ILR6 Register 194
2-45. INTCPS_ILR7 Register 195
2-46. INTCPS_ILR8 Register 196
2-47. INTCPS_ILR9 Register 197

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2-48. INTCPS_ILR10 Register 198
2-49. INTCPS_ILR11 Register 199
2-50. INTCPS_ILR12 Register 200
2-51. INTCPS_ILR13 Register 201
2-52. INTCPS_ILR14 Register 202
2-53. INTCPS_ILR15 Register 203
2-54. INTCPS_ILR16 Register 204
2-55. INTCPS_ILR17 Register 205
2-56. INTCPS_ILR18 Register 206
2-57. INTCPS_ILR19 Register 207
2-58. INTCPS_ILR20 Register 208
2-59. INTCPS_ILR21 Register 209
2-60. INTCPS_ILR22 Register 210
2-61. INTCPS_ILR23 Register 211
2-62. INTCPS_ILR24 Register 212
2-63. INTCPS_ILR25 Register 213
2-64. INTCPS_ILR26 Register 214
2-65. INTCPS_ILR27 Register 215
2-66. INTCPS_ILR28 Register 216
2-67. INTCPS_ILR29 Register 217
2-68. INTCPS_ILR30 Register 218
2-69. INTCPS_ILR31 Register 219
2-70. INTCPS_ILR32 Register 220
2-71. INTCPS_ILR33 Register 221
2-72. INTCPS_ILR34 Register 222
2-73. INTCPS_ILR35 Register 223
2-74. INTCPS_ILR36 Register 224
2-75. INTCPS_ILR37 Register 225
2-76. INTCPS_ILR38 Register 226
2-77. INTCPS_ILR39 Register 227
2-78. INTCPS_ILR40 Register 228
2-79. INTCPS_ILR41 Register 229
2-80. INTCPS_ILR42 Register 230
2-81. INTCPS_ILR43 Register 231
2-82. INTCPS_ILR44 Register 232
2-83. INTCPS_ILR45 Register 233
2-84. INTCPS_ILR46 Register 234
2-85. INTCPS_ILR47 Register 235
2-86. INTCPS_ILR48 Register 236
2-87. INTCPS_ILR49 Register 237
2-88. INTCPS_ILR50 Register 238
2-89. INTCPS_ILR51 Register 239
2-90. INTCPS_ILR52 Register 240
2-91. INTCPS_ILR53 Register 241
2-92. INTCPS_ILR54 Register 242
2-93. INTCPS_ILR55 Register 243
2-94. INTCPS_ILR56 Register 244
2-95. INTCPS_ILR57 Register 245
2-96. INTCPS_ILR58 Register 246

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2-97. INTCPS_ILR59 Register 247
2-98. INTCPS_ILR60 Register 248
2-99. INTCPS_ILR61 Register 249
2-100. INTCPS_ILR62 Register 250
2-101. INTCPS_ILR63 Register 251
2-102. INTCPS_ILR64 Register 252
2-103. INTCPS_ILR65 Register 253
2-104. INTCPS_ILR66 Register 254
2-105. INTCPS_ILR67 Register 255
2-106. INTCPS_ILR68 Register 256
2-107. INTCPS_ILR69 Register 257
2-108. INTCPS_ILR70 Register 258
2-109. INTCPS_ILR71 Register 259
2-110. INTCPS_ILR72 Register 260
2-111. INTCPS_ILR73 Register 261
2-112. INTCPS_ILR74 Register 262
2-113. INTCPS_ILR75 Register 263
2-114. INTCPS_ILR76 Register 264
2-115. INTCPS_ILR77 Register 265
2-116. INTCPS_ILR78 Register 266
2-117. INTCPS_ILR79 Register 267
2-118. INTCPS_ILR80 Register 268
2-119. INTCPS_ILR81 Register 269
2-120. INTCPS_ILR82 Register 270
2-121. INTCPS_ILR83 Register 271
2-122. INTCPS_ILR84 Register 272
2-123. INTCPS_ILR85 Register 273
2-124. INTCPS_ILR86 Register 274
2-125. INTCPS_ILR87 Register 275
2-126. INTCPS_ILR88 Register 276
2-127. INTCPS_ILR89 Register 277
2-128. INTCPS_ILR90 Register 278
2-129. INTCPS_ILR91 Register 279
2-130. INTCPS_ILR92 Register 280
2-131. INTCPS_ILR93 Register 281
2-132. INTCPS_ILR94 Register 282
2-133. INTCPS_ILR95 Register 283
2-134. INTCPS_SIR_IRQ Register 284
2-135. INTCPS_ILR96 Register 285
2-136. INTCPS_ILR97 Register 286
2-137. INTCPS_ILR98 Register 287
2-138. INTCPS_ILR99 Register 288
2-139. INTCPS_ILR100 Register 289
2-140. INTCPS_ILR101 Register 290
2-141. INTCPS_ILR102 Register 291
2-142. INTCPS_ILR103 Register 292
2-143. INTCPS_ILR104 Register 293
2-144. INTCPS_ILR105 Register 294
2-145. INTCPS_SIR_FIQ Register 295

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2-146. INTCPS_ILR106 Register 296
2-147. INTCPS_ILR107 Register 297
2-148. INTCPS_ILR108 Register 298
2-149. INTCPS_ILR109 Register 299
2-150. INTCPS_ILR110 Register 300
2-151. INTCPS_ILR111 Register 301
2-152. INTCPS_ILR112 Register 302
2-153. INTCPS_ILR113 Register 303
2-154. INTCPS_ILR114 Register 304
2-155. INTCPS_ILR115 Register 305
2-156. INTCPS_CONTROL Register 306
2-157. INTCPS_ILR116 Register 307
2-158. INTCPS_ILR117 Register 308
2-159. INTCPS_ILR118 Register 309
2-160. INTCPS_ILR119 Register 310
2-161. INTCPS_ILR120 Register 311
2-162. INTCPS_ILR121 Register 312
2-163. INTCPS_ILR122 Register 313
2-164. INTCPS_ILR123 Register 314
2-165. INTCPS_ILR124 Register 315
2-166. INTCPS_ILR125 Register 316
2-167. INTCPS_PROTECTION Register 317
2-168. INTCPS_ILR126 Register 318
2-169. INTCPS_ILR127 Register 319
2-170. INTCPS_IDLE Register 320
2-171. INTCPS_IRQ_PRIORITY Register 321
4-1. Block Diagram 338
4-2. REVID Register 342
4-3. CMR0 Register 343
4-4. CMR1 Register 344
4-5. CMR2 Register 345
4-6. CMR3 Register 346
4-7. CMR4 Register 347
4-8. CMR5 Register 348
4-9. CMR6 Register 349
4-10. CMR7 Register 350
4-11. CMR8 Register 351
4-12. CMR9 Register 352
4-13. CMR10 Register 353
4-14. CMR11 Register 354
4-15. CMR12 Register 355
4-16. CMR13 Register 356
4-17. CMR14 Register 357
4-18. CMR15 Register 358
4-19. GPIR Register 359
4-20. GER Register 360
4-21. HMR0 Register 361
4-22. HMR1 Register 362
4-23. HMR2 Register 363

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4-24. HIPIR0 Register 364
4-25. HIPIR1 Register 365
4-26. HIPIR2 Register 366
4-27. HIPIR3 Register 367
4-28. HIPIR4 Register 368
4-29. HIPIR5 Register 369
4-30. HIPIR6 Register 370
4-31. HIPIR7 Register 371
4-32. HIPIR8 Register 372
4-33. HIPIR9 Register 373
4-34. GNLR Register 374
4-35. SISR Register 375
4-36. SIPR0 Register 376
4-37. SIPR1 Register 377
4-38. SITR0 Register 378
4-39. SITR1 Register 379
4-40. SICR Register 380
4-41. CR Register 381
4-42. EISR Register 382
4-43. HINLR0 Register 383
4-44. HINLR1 Register 384
4-45. HINLR2 Register 385
4-46. HINLR3 Register 386
4-47. HINLR4 Register 387
4-48. HINLR5 Register 388
4-49. HINLR6 Register 389
4-50. HINLR7 Register 390
4-51. HINLR8 Register 391
4-52. HINLR9 Register 392
4-53. EICR Register 393
4-54. SRSR0 Register 394
4-55. SRSR1 Register 395
4-56. HIEISR Register 396
4-57. HIER Register 397
4-58. HIDISR Register 398
4-59. SECR0 Register 399
4-60. SECR1 Register 400
4-61. ESR0 Register 401
4-62. ERS1 Register 402
4-63. ECR0 Register 403
4-64. ECR1 Register 404
4-65. CONTROL Register 405
4-66. CYCLE Register 406
4-67. STALL Register 407
4-68. CTBIR0 Register 408
4-69. CTBIR1 Register 409
4-70. STATUS Register 410
4-71. CTPPR0 Register 411
4-72. CTPPR1 Register 412

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4-73. WAKEUP_EN Register 413
4-74. GPR0 Register 415
4-75. GPR25 Register 416
4-76. GPR26 Register 417
4-77. GPR27 Register 418
4-78. GPR28 Register 419
4-79. GPR29 Register 420
4-80. GPR3 Register 421
4-81. GPR30 Register 422
4-82. GPR31 Register 423
4-83. CTR0 Register 424
4-84. CTR1 Register 425
4-85. CTR2 Register 426
4-86. CTR3 Register 427
4-87. CTR4 Register 428
4-88. CTR5 Register 429
4-89. CTR6 Register 430
4-90. CTR7 Register 431
4-91. GPR4 Register 432
4-92. CTR8 Register 433
4-93. CTR9 Register 434
4-94. CTR10 Register 435
4-95. CTR11 Register 436
4-96. CTR12 Register 437
4-97. CTR13 Register 438
4-98. CTR14 Register 439
4-99. CTR15 Register 440
4-100. CTR16 Register 441
4-101. CTR17 Register 442
4-102. GPR5 Register 443
4-103. CTR18 Register 444
4-104. CTR19 Register 445
4-105. CTR20 Register 446
4-106. CTR21 Register 447
4-107. CTR22 Register 448
4-108. CTR23 Register 449
4-109. CTR24 Register 450
4-110. GPR6 Register 451
4-111. GPR7 Register 452
4-112. GPR8 Register 453
4-113. GPR9 Register 454
4-114. GPR1 Register 455
4-115. GPR10 Register 456
4-116. GPR11 Register 457
4-117. GPR12 Register 458
4-118. GPR13 Register 459
4-119. GPR14 Register 460
4-120. GPR15 Register 461
4-121. GPR16 Register 462

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4-122. GPR17 Register 463
4-123. GPR18 Register 464
4-124. GPR19 Register 465
4-125. GPR2 Register 466
4-126. GPR20 Register 467
4-127. GPR21 Register 468
4-128. GPR22 Register 469
4-129. GPR23 Register 470
4-130. GPR24 Register 471
4-131. REVID Register 472
4-132. GPCFG1 Register 473
4-133. CGR Register 474
4-134. ISRP Register 475
4-135. ISP Register 476
4-136. IESP Register 477
4-137. IECP Register 478
4-138. SCRP Register 479
4-139. SYSCFG Register 480
4-140. PMAO Register 481
4-141. MII_RT Register 482
4-142. IEPCLK Register 483
4-143. SPP Register 484
4-144. PIN_MX Register 485
4-145. GPCFG0 Register 486
4-146. UART RBR/TBR Registers Register 488
4-147. UART Line Control Register Register 489
4-148. UART Modem Control Register Register 490
4-149. UART Line Status Register Register 491
4-150. UART Modem Status Register Register 492
4-151. UART Scratch Register Register 493
4-152. UART Divisor Register (LSB) Register 494
4-153. UART Divisor Register (MSB) Register 495
4-154. UART Interrupt Enable Register Register 496
4-155. UART Peripheral ID Register Register 497
4-156. UART PowerManagement and Emulation Register Register 498
4-157. UART Mode Definition Register Register 499
4-158. UART Interrupt Identification Register / FIFO Control Register Register 500
4-159. GLOBAL_CFG Register 502
4-160. CMP7 Register 503
4-161. COUNT Register 504
4-162. RXIPG0 Register 505
4-163. RXIPG1 Register 506
4-164. CAP_CFG Register 507
4-165. CAP_STATUS Register 508
4-166. CAPR0 Register 509
4-167. SYNC_CTRL Register 510
4-168. SYNC_FIRST_STAT Register 511
4-169. SYNC0_STAT Register 512
4-170. SYNC1_STAT Register 513

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4-171. SYNC_PWIDTH Register 514
4-172. SYNC0_PERIOD Register 515
4-173. CAPR1 Register 516
4-174. SYNC1_DELAY Register 517
4-175. SYNC_START Register 518
4-176. CAPR2 Register 519
4-177. CAPR3 Register 520
4-178. GLOBAL_STATUS Register 521
4-179. CAPR4 Register 522
4-180. CAPR5 Register 523
4-181. CAPR6 Register 524
4-182. WD_PREDIV Register 525
4-183. PDI_WD_TIM Register 526
4-184. CAPF6 Register 527
4-185. PD_WD_TIM Register 528
4-186. WD_STATUS Register 529
4-187. WD_EXP_CNT Register 530
4-188. WD_CTRL Register 531
4-189. CAPR7 Register 532
4-190. CAPF7 Register 533
4-191. CMP_CFG Register 534
4-192. CMP_STATUS Register 535
4-193. CMP0 Register 536
4-194. CMP1 Register 537
4-195. DIGIO_CTRL Register 538
4-196. DIGIO_STATUS Register 539
4-197. DIGIO_DATA_IN Register 540
4-198. DIGIO_DATA_IN_RAW Register 541
4-199. DIGIO_DATA_OUT Register 542
4-200. DIGIO_DATA_OUT_EN Register 543
4-201. DIGIO_EXP Register 544
4-202. COMPEN Register 545
4-203. CMP2 Register 546
4-204. CMP3 Register 547
4-205. CMP4 Register 548
4-206. CMP5 Register 549
4-207. CMP6 Register 550
4-208. TSCNT Register 551
4-209. CAP2 Register 552
4-210. CAP3 Register 553
4-211. CAP4 Register 554
4-212. ECCTL1 Register 555
4-213. CNTPHS Register 557
4-214. ECCTL2 Register 558
4-215. ECEINT Register 560
4-216. ECFLG Register 561
4-217. ECCLR Register 562
4-218. ECFRC Register 563
4-219. CAP1 Register 564

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4-220. PID Register 565
4-221. RXCFG0 Register 566
4-222. TXCFG0 Register 567
4-223. TXCFG1 Register 568
4-224. TX_CRC0 Register 569
4-225. TX_CRC1 Register 570
4-226. RXCFG1 Register 571
4-227. TX_IPG0 Register 572
4-228. TX_IPG1 Register 573
4-229. PRS0 Register 574
4-230. PRS1 Register 575
4-231. RX_FRMS0 Register 576
4-232. RX_FRMS1 Register 577
4-233. RX_PCNT0 Register 578
4-234. RX_PCNT1 Register 579
4-235. RX_ERR0 Register 580
4-236. RX_ERR1 Register 581
4-237. VER Register 583
4-238. LINK Register 584
4-239. USERACCESS0 Register 585
4-240. USERPHYSEL0 Register 586
4-241. USERACCESS1 Register 587
4-242. USERPHYSEL1 Register 588
4-243. LINKINTRAW Register 589
4-244. LINKINTMASKED Register 590
4-245. USERINTRAW Register 591
4-246. USERINTMASKED Register 592
4-247. CONTROL Register 593
4-248. USERINTMASKSET Register 594
4-249. USERINTMASKCLR Register 595
4-250. ALIVE Register 596
5-1. EUR_CR_CLKGATECTL Register 609
5-2. EUR_CR_PERF_COUNTER8 Register 611
5-3. EUR_CR_PIXELBE Register 612
5-4. EUR_CR_ISP_RENDER Register 613
5-5. EUR_CR_ISP_RGN_BASE Register 614
5-6. EUR_CR_ISP_RENDBOX1 Register 615
5-7. EUR_CR_ISP_RENDBOX2 Register 616
5-8. EUR_CR_ISP_IPFMISC Register 617
5-9. EUR_CR_ISP_OGL_MODE Register 618
5-10. EUR_CR_ISP_PERPENDICULAR Register 619
5-11. EUR_CR_ISP_CULLVALUE Register 620
5-12. EUR_CR_ISP_DBIAS Register 621
5-13. EUR_CR_ISP_START_RENDER Register 622
5-14. EUR_CR_3D_AA_MODE Register 623
5-15. EUR_CR_ISP_BREAK Register 624
5-16. EUR_CR_ISP_3DCONTEXT Register 625
5-17. EUR_CR_ISP_FPU Register 626
5-18. EUR_CR_TSP_PARAMETER_CACHE Register 627

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5-19. EUR_CR_TSP_PARAMETER_FETCH Register 628
5-20. EUR_CR_ISP_ZLSCTL Register 629
5-21. EUR_CR_ISP_ZLOAD_BASE Register 630
5-22. EUR_CR_ISP_ZSTORE_BASE Register 631
5-23. EUR_CR_ISP_STENCIL_LOAD_BASE Register 632
5-24. EUR_CR_ISP_STENCIL_STORE_BASE Register 633
5-25. EUR_CR_ISP_VISTEST_CTRL Register 634
5-26. EUR_CR_ISP_VISTEST_VISIBLE0 Register 635
5-27. EUR_CR_ISP_VISTEST_VISIBLE1 Register 636
5-28. EUR_CR_ISP_VISTEST_VISIBLE2 Register 637
5-29. EUR_CR_ISP_VISTEST_VISIBLE3 Register 638
5-30. EUR_CR_ISP_VISTEST_VISIBLE4 Register 639
5-31. EUR_CR_ISP_VISTEST_VISIBLE5 Register 640
5-32. EUR_CR_ISP_VISTEST_VISIBLE6 Register 641
5-33. EUR_CR_ISP_VISTEST_VISIBLE7 Register 642
5-34. EUR_CR_ISP_BGOBJDEPTH Register 643
5-35. EUR_CR_ISP_BGOBJ Register 644
5-36. EUR_CR_ISP_BGOBJTAG Register 645
5-37. EUR_CR_ISP_MULTISAMPLECTL Register 646
5-38. EUR_CR_ISP_SIG1 Register 647
5-39. EUR_CR_ISP_SIG2 Register 648
5-40. EUR_CR_ISP_SIG3 Register 649
5-41. EUR_CR_ISP_SIG4 Register 650
5-42. EUR_CR_ISP_TAGCTRL Register 651
5-43. EUR_CR_ISP_STATUS1 Register 652
5-44. EUR_CR_ISP_STATUS2 Register 653
5-45. EUR_CR_ISP_DEPTHSORT Register 654
5-46. EUR_CR_PIXELBE_EMIT Register 655
5-47. EUR_CR_SOFT_RESET Register 656
5-48. EUR_CR_DPM_3D_PAGE_TABLE_BASE Register 657
5-49. EUR_CR_DPM_3D_FREE_LIST Register 658
5-50. EUR_CR_DPM_PDS_PAGE_THRESHOLD Register 659
5-51. EUR_CR_DPM_TA_ALLOC_PAGE_TABLE_BASE Register 660
5-52. EUR_CR_DPM_TA_ALLOC_FREE_LIST Register 661
5-53. EUR_CR_DPM_TA_PAGE_THRESHOLD Register 662
5-54. EUR_CR_DPM_ZLS_PAGE_THRESHOLD Register 663
5-55. EUR_CR_DPM_TA_GLOBAL_LIST Register 664
5-56. EUR_CR_DPM_STATE_TABLE_BASE Register 665
5-57. EUR_CR_DPM_STATE_CONTEXT_ID Register 666
5-58. EUR_CR_DPM_CONTROL_TABLE_BASE Register 667
5-59. EUR_CR_DPM_START_OF_CONTEXT_PAGES_ALLOCATED Register 668
5-60. EUR_CR_DPM_3D_DEALLOCATE Register 669
5-61. EUR_CR_CORE_ID Register 670
5-62. EUR_CR_DPM_ALLOC Register 671
5-63. EUR_CR_DPM_DALLOC Register 672
5-64. EUR_CR_DPM_TA_ALLOC Register 673
5-65. EUR_CR_DPM_3D Register 674
5-66. EUR_CR_DPM_PARTIAL_RENDER Register 675
5-67. EUR_CR_DPM_LSS_PARTIAL_CONTEXT Register 676

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5-68. EUR_CR_DPM_START_OF_3D_CONTEXT_PAGES_ALLOCATED Register 677
5-69. EUR_CR_DPM_TASK_3D_FREE Register 678
5-70. EUR_CR_DPM_TASK_TA_FREE Register 679
5-71. EUR_CR_DPM_TASK_STATE Register 680
5-72. EUR_CR_DPM_TASK_CONTROL Register 681
5-73. EUR_CR_DPM_OUTOFMEM Register 682
5-74. EUR_CR_DPM_FREE_CONTEXT Register 683
5-75. EUR_CR_DPM_3D_TIMEOUT Register 684
5-76. EUR_CR_DPM_TA_EVM Register 685
5-77. EUR_CR_DPM_TA_ALLOC_FREE_LIST_STATUS1 Register 686
5-78. EUR_CR_DPM_3D_FREE_LIST_STATUS1 Register 687
5-79. EUR_CR_DPM_TA_ALLOC_FREE_LIST_STATUS2 Register 688
5-80. EUR_CR_DPM_3D_FREE_LIST_STATUS2 Register 689
5-81. EUR_CR_DPM_ABORT_STATUS_MTILE Register 690
5-82. EUR_CR_DPM_PAGE_STATUS Register 691
5-83. EUR_CR_DPM_PAGE Register 692
5-84. EUR_CR_DPM_GLOBAL_PAGE_STATUS Register 693
5-85. EUR_CR_DPM_GLOBAL_PAGE Register 694
5-86. EUR_CR_DPM_REQUESTING Register 695
5-87. EUR_CR_DPM_RNG_HDR_PARSER Register 696
5-88. EUR_CR_CORE_REVISION Register 697
5-89. EUR_CR_TSP_CONFIG Register 698
5-90. EUR_CR_CACHE_CTRL Register 699
5-91. EUR_CR_DESIGNER_REV_FIELD1 Register 700
5-92. EUR_CR_TRIGGER Register 701
5-93. EUR_CR_USE_CTRL Register 702
5-94. EUR_CR_USE_LD_REDIRECT Register 703
5-95. EUR_CR_USE_CACHE Register 704
5-96. EUR_CR_USE_CODE_BASE00 Register 705
5-97. EUR_CR_USE_CODE_BASE01 Register 706
5-98. EUR_CR_USE_CODE_BASE02 Register 707
5-99. EUR_CR_USE_CODE_BASE03 Register 708
5-100. EUR_CR_USE_CODE_BASE04 Register 709
5-101. EUR_CR_USE_CODE_BASE05 Register 710
5-102. EUR_CR_USE_CODE_BASE06 Register 711
5-103. EUR_CR_USE_CODE_BASE07 Register 712
5-104. EUR_CR_USE_CODE_BASE08 Register 713
5-105. EUR_CR_USE_CODE_BASE09 Register 714
5-106. EUR_CR_USE_CODE_BASE10 Register 715
5-107. EUR_CR_USE_CODE_BASE11 Register 716
5-108. EUR_CR_USE_CODE_BASE12 Register 717
5-109. EUR_CR_USE_CODE_BASE13 Register 718
5-110. EUR_CR_USE_CODE_BASE14 Register 719
5-111. EUR_CR_USE_CODE_BASE15 Register 720
5-112. EUR_CR_EVENT_PDS_ENABLE2 Register 721
5-113. EUR_CR_PDS_CACHE_ENABLE Register 722
5-114. EUR_CR_EVENT_PDS_ENABLE Register 723
5-115. EUR_CR_EVENT_PIXEL_PDS_EXEC Register 725
5-116. EUR_CR_EVENT_PIXEL_PDS_DATA Register 726

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5-117. EUR_CR_EVENT_PIXEL_PDS_INFO Register 727
5-118. EUR_CR_EVENT_OTHER_PDS_EXEC Register 728
5-119. EUR_CR_EVENT_OTHER_PDS_DATA Register 729
5-120. EUR_CR_EVENT_OTHER_PDS_INFO Register 730
5-121. EUR_CR_DMS_CTRL Register 731
5-122. EUR_CR_DMS_AGE Register 732
5-123. EUR_CR_USE_G0 Register 733
5-124. EUR_CR_USE_G1 Register 734
5-125. EUR_CR_USE_FILTER0_LEFT Register 735
5-126. EUR_CR_USE_FILTER0_RIGHT Register 736
5-127. EUR_CR_USE_FILTER0_EXTRA Register 737
5-128. EUR_CR_USE_FILTER1_LEFT Register 738
5-129. EUR_CR_USE_FILTER1_RIGHT Register 739
5-130. EUR_CR_USE_FILTER1_EXTRA Register 740
5-131. EUR_CR_USE_FILTER_TABLE Register 741
5-132. EUR_CR_EVENT_HOST_ENABLE2 Register 742
5-133. EUR_CR_USE_ST_RANGE Register 743
5-134. EUR_CR_USE0_DM_SLOT Register 744
5-135. EUR_CR_USE1_DM_SLOT Register 746
5-136. EUR_CR_USE_EXEF Register 748
5-137. EUR_CR_PDS_EXEC_BASE Register 749
5-138. EUR_CR_PDS Register 750
5-139. EUR_CR_EVENT_KICKER Register 751
5-140. EUR_CR_EVENT_HOST_CLEAR2 Register 752
5-141. EUR_CR_EVENT_KICK Register 753
5-142. EUR_CR_EVENT_TIMER Register 754
5-143. EUR_CR_PDS_INV0 Register 755
5-144. EUR_CR_PDS_INV1 Register 756
5-145. EUR_CR_PDS_INV2 Register 757
5-146. EUR_CR_PDS_INV3 Register 758
5-147. EUR_CR_PDS_INV_CSC Register 759
5-148. EUR_CR_USE_GLOBCOM0 Register 760
5-149. EUR_CR_USE_GLOBCOM1 Register 761
5-150. EUR_CR_USE_GLOBCOM2 Register 762
5-151. EUR_CR_DESIGNER_REV_FIELD2 Register 763
5-152. EUR_CR_EVENT_STATUS2 Register 764
5-153. EUR_CR_USE_GLOBCOM3 Register 765
5-154. EUR_CR_USE_GLOBCOM4 Register 766
5-155. EUR_CR_USE_GLOBCOM5 Register 767
5-156. EUR_CR_USE_GLOBCOM6 Register 768
5-157. EUR_CR_USE_GLOBCOM7 Register 769
5-158. EUR_CR_LOOPBACK Register 770
5-159. EUR_CR_USE0_SERV_PIXEL Register 771
5-160. EUR_CR_USE0_SERV_VERTEX Register 772
5-161. EUR_CR_USE0_SERV_EVENT Register 773
5-162. EUR_CR_USE1_SERV_PIXEL Register 774
5-163. EUR_CR_USE1_SERV_VERTEX Register 775
5-164. EUR_CR_USE1_SERV_EVENT Register 776
5-165. EUR_CR_USE_FILTER2_LEFT Register 777

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5-166. EUR_CR_USE_FILTER2_RIGHT Register 778
5-167. EUR_CR_USE_FILTER2_EXTRA Register 779
5-168. EUR_CR_PDS_PC_BASE Register 780
5-169. EUR_CR_USE_MOE Register 781
5-170. EUR_CR_EVENT_STATUS Register 782
5-171. EUR_CR_EVENT_HOST_ENABLE Register 784
5-172. EUR_CR_BIF_CTRL Register 786
5-173. EUR_CR_BIF_INT_STAT Register 787
5-174. EUR_CR_EVENT_HOST_CLEAR Register 788
5-175. EUR_CR_BIF_FAULT Register 789
5-176. EUR_CR_PDS_CACHE_STATUS Register 790
5-177. EUR_CR_PDS_CACHE_HOST_ENABLE Register 791
5-178. EUR_CR_PDS_CACHE_HOST_CLEAR Register 792
5-179. EUR_CR_BIF_DIR_LIST_BASE0 Register 793
5-180. EUR_CR_BIF_TA_REQ_BASE Register 794
5-181. EUR_CR_BIF_MEM_REQ_STAT Register 795
5-182. EUR_CR_BIF_3D_REQ_BASE Register 796
5-183. EUR_CR_BIF_ZLS_REQ_BASE Register 797
5-184. EUR_CR_CLKGATESTATUS Register 798
5-185. EUR_CR_VDM_START Register 799
5-186. EUR_CR_TE_AA Register 800
5-187. EUR_CR_TE_MTILE1 Register 801
5-188. EUR_CR_TE_MTILE2 Register 802
5-189. EUR_CR_TE_SCREEN Register 803
5-190. EUR_CR_TE_MTILE Register 804
5-191. EUR_CR_TE_PSG Register 805
5-192. EUR_CR_TE_PSGREGION_BASE Register 806
5-193. EUR_CR_TE_TPC_BASE Register 807
5-194. EUR_CR_TE_TPCCONTROL Register 808
5-195. EUR_CR_TE_RGNBBOX_X Register 809
5-196. EUR_CR_TE_RGNBBOX_Y Register 810
5-197. EUR_CR_MTE_OTPM_CSM_FLUSH_BASE Register 811
5-198. EUR_CR_MTE_OTPM_CSM_LOAD_BASE Register 812
5-199. EUR_CR_VDM_CTRL_STREAM_BASE Register 813
5-200. EUR_CR_MTE_CTRL Register 814
5-201. EUR_CR_MTE_WCOMPARE Register 815
5-202. EUR_CR_MTE_WCLAMP Register 816
5-203. EUR_CR_MTE_SCREEN Register 817
5-204. EUR_CR_MTE_OTPM_OP Register 818
5-205. EUR_CR_MTE_MULTISAMPLECTL Register 819
5-206. EUR_CR_MTE_SIG1 Register 820
5-207. EUR_CR_MTE_SIG2 Register 821
5-208. EUR_CR_TE_STATE Register 822
5-209. EUR_CR_PERF Register 823
5-210. EUR_CR_PERF_COUNTER0 Register 824
5-211. EUR_CR_PERF_COUNTER1 Register 825
5-212. EUR_CR_PERF_COUNTER2 Register 826
5-213. EUR_CR_CLIP_SIG1 Register 827
5-214. EUR_CR_PIXELBE_SIG01 Register 828

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5-215. EUR_CR_PIXELBE_SIG02 Register 829
5-216. EUR_CR_CLKGATECTLOVR Register 830
5-217. EUR_CR_PERF_COUNTER3 Register 831
5-218. EUR_CR_VDM_CONTEXT_STORE_START Register 832
5-219. EUR_CR_VDM_CONTEXT_STORE_STREAM Register 833
5-220. EUR_CR_VDM_CONTEXT_STORE_INDEX Register 834
5-221. EUR_CR_VDM_CONTEXT_STORE_STATUS Register 835
5-222. EUR_CR_VDM_CONTEXT_STORE_STATE0 Register 836
5-223. EUR_CR_VDM_CONTEXT_STORE_STATE1 Register 837
5-224. EUR_CR_PERF_COUNTER4 Register 838
5-225. EUR_CR_VDM_WAIT_FOR_KICK Register 839
5-226. EUR_CR_VDM_CONTEXT_STORE_WRAP_INDEX Register 840
5-227. EUR_CR_VDM_CONTEXT_STORE_WRAPPED Register 841
5-228. EUR_CR_VDM_CONTEXT_STORE_FAN Register 842
5-229. EUR_CR_VDM_CONTEXT_STORE_WRAP Register 843
5-230. EUR_CR_PERF_COUNTER5 Register 844
5-231. EUR_CR_PERF_COUNTER6 Register 845
5-232. EUR_CR_PERF_COUNTER7 Register 846
6-1. GPMC Block Diagram 850
6-2. GPMC Integration 852
6-3. GPMC to 16-Bit Address/Data-Multiplexed Memory 856
6-4. GPMC to 16-Bit Nonmultiplexed Memory 857
6-5. GPMC to 8-Bit NAND Device 857
6-6. GPMC Integration 858
6-7. Chip-Select Address Mapping and Decoding Mask 864
6-8. Wait Behavior During an Asynchronous Single Read Access (GPMCFCLKDivider = 1) 867
6-9. Wait Behavior During a Synchronous Read Burst Access 869
6-10. Read to Read for an Address-Data Multiplexed Device, On Different CS, Without Bus Turnaround ($\overline{CS0}$ Attached to Fast Device) 871
6-11. Read to Read / Write for an Address-Data Multiplexed Device, On Different CS, With Bus Turnaround 871
6-12. Read to Read / Write for a Address-Data or AAD-Multiplexed Device, On Same CS, With Bus Turnaround 872
6-13. Asynchronous Single Read Operation on an Address/Data Multiplexed Device 881
6-14. Two Asynchronous Single Read Accesses on an Address/Data Multiplexed Device (32-Bit Read Split Into 2 × 16-Bit Read) 882
6-15. Asynchronous Single Write on an Address/Data-Multiplexed Device 883
6-16. Asynchronous Single-Read on an AAD-Multiplexed Device 884
6-17. Asynchronous Single Write on an AAD-Multiplexed Device 886
6-18. Synchronous Single Read (GPMCFCLKDIVIDER = 0) 888
6-19. Synchronous Single Read (GPMCFCLKDIVIDER = 1) 889
6-20. Synchronous Multiple (Burst) Read (GPMCFCLKDIVIDER = 0) 891
6-21. Synchronous Multiple (Burst) Read (GPMCFCLKDIVIDER = 1) 892
6-22. Synchronous Single Write on an Address/Data-Multiplexed Device 893
6-23. Synchronous Multiple Write (Burst Write) in Address/Data-Multiplexed Mode 894
6-24. Synchronous Multiple Write (Burst Write) in Address/Address/Data-Multiplexed Mode 895
6-25. Asynchronous Single Read on an Address/Data-Nonmultiplexed Device 897
6-26. Asynchronous Single Write on an Address/Data-Nonmultiplexed Device 898
6-27. Asynchronous Multiple (Page Mode) Read 899
6-28. NAND Command Latch Cycle 904
6-29. NAND Address Latch Cycle 905

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6-30. NAND Data Read Cycle 906
6-31. NAND Data Write Cycle 907
6-32. Hamming Code Accumulation Algorithm (1 of 2) 911
6-33. Hamming Code Accumulation Algorithm (2 of 2) 912
6-34. ECC Computation for a 256-Byte Data Stream (Read or Write) 912
6-35. ECC Computation for a 512-Byte Data Stream (Read or Write) 913
6-36. 128 Word16 ECC Computation 914
6-37. 256 Word16 ECC Computation 914
6-38. Manual Mode Sequence and Mapping 919
6-39. NAND Page Mapping and ECC: Per-Sector Schemes 924
6-40. NAND Page Mapping and ECC: Pooled Spare Schemes 925
6-41. NAND Page Mapping and ECC: Per-Sector Schemes, with Separate ECC 926
6-42. NAND Read Cycle Optimization Timing Description 933
6-43. Programming Model Top-Level Diagram 936
6-44. NOR Interfacing Timing Parameters Diagram 943
6-45. NAND Command Latch Cycle Timing Simplified Example 947
6-46. Synchronous NOR Single Read Simplified Example 952
6-47. Asynchronous NOR Single Write Simplified Example 954
6-48. GPMC Connection to an External NOR Flash Memory 956
6-49. Synchronous Burst Read Access (Timing Parameters in Clock Cycles) 958
6-50. Asynchronous Single Read Access (Timing Parameters in Clock Cycles) 960
6-51. Asynchronous Single Write Access (Timing Parameters in Clock Cycles) 962
6-52. GPMC_REVISION 968
6-53. GPMC_SYSCONFIG 968
6-54. GPMC_SYSSTATUS 969
6-55. GPMC_IRQSTATUS 970
6-56. GPMC_IRQENABLE 971
6-57. GPMC_TIMEOUT_CONTROL 972
6-58. GPMC_ERR_ADDRESS 972
6-59. GPMC_ERR_TYPE 973
6-60. GPMC_CONFIG 974
6-61. GPMC_STATUS 975
6-62. GPMC_CONFIG1_i 976
6-63. GPMC_CONFIG2_i 979
6-64. GPMC_CONFIG3_i 980
6-65. GPMC_CONFIG4_i 982
6-66. GPMC_CONFIG5_i 984
6-67. GPMC_CONFIG6_i 985
6-68. GPMC_CONFIG7_i 986
6-69. GPMC_NAND_COMMAND_i 987
6-70. GPMC_NAND_ADDRESS_i 987
6-71. GPMC_NAND_DATA_i 987
6-72. GPMC_PREFETCH_CONFIG1 988
6-73. GPMC_PREFETCH_CONFIG2 990
6-74. GPMC_PREFETCH_CONTROL 990
6-75. GPMC_PREFETCH_STATUS 991
6-76. GPMC_ECC_CONFIG 992
6-77. GPMC_ECC_CONTROL 993
6-78. GPMC_ECC_SIZE_CONFIG 994

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6-79. GPMC_ECCj_RESULT 996
6-80. GPMC_BCH_RESULT0_i 997
6-81. GPMC_BCH_RESULT1_i 997
6-82. GPMC_BCH_RESULT2_i 997
6-83. GPMC_BCH_RESULT3_i 998
6-84. GPMC_BCH_SWDATA 998
6-85. GPMC_BCH_RESULT4_i 998
6-86. GPMC_BCH_RESULT5_i 999
6-87. GPMC_BCH_RESULT6_i 999
6-88. OCMC RAM Integration 1001
6-89. EMIF_MOD_ID_REV Register 1009
6-90. IODFT_CTRL_MISR_RSLT Register 1010
6-91. IODFT_ADDR_MISR_RSLT Register 1011
6-92. IODFT_DATA_MISR_RSLT_1 Register 1012
6-93. IODFT_DATA_MISR_RSLT_2 Register 1013
6-94. IODFT_DATA_MISR_RSLT_3 Register 1014
6-95. SDRAM_CONFIG_2 Register 1015
6-96. PERF_CNT_1 Register 1016
6-97. PERF_CNT_2 Register 1017
6-98. PERF_CNT_CFG Register 1018
6-99. PERF_CNT_SEL Register 1019
6-100. PERF_CNT_TIM Register 1020
6-101. READ_IDLE_CTRL Register 1021
6-102. READ_IDLE_CTRL_SHDW Register 1022
6-103. SDRAM_REF_CTRL Register 1023
6-104. IRQ_EOI Register 1024
6-105. IRQSTATUS_RAW_SYS Register 1025
6-106. IRQSTATUS_RAW_LL Register 1026
6-107. IRQSTATUS_SYS Register 1027
6-108. IRQSTATUS_LL Register 1028
6-109. IRQENABLE_SET_SYS Register 1029
6-110. IRQENABLE_SET_LL Register 1030
6-111. IRQENABLE_CLR_SYS Register 1031
6-112. IRQENABLE_CLR_LL Register 1032
6-113. SDRAM_REF_CTRL_SHDW Register 1033
6-114. ZQ_CONFIG Register 1034
6-115. TEMP_ALERT_CONFIG Register 1035
6-116. OCP_ERR_LOG Register 1036
6-117. DDR_PHY_CTRL_1 Register 1037
6-118. DDR_PHY_CTRL_1_SHDW Register 1038
6-119. DDR_PHY_CTRL_2 Register 1039
6-120. SDRAM_TIM_1 Register 1040
6-121. SDRAM_TIM_1_SHDW Register 1041
6-122. SDRAM_TIM_2 Register 1042
6-123. SDRAM_TIM_2_SHDW Register 1043
6-124. STATUS Register 1044
6-125. SDRAM_TIM_3 Register 1045
6-126. SDRAM_TIM_3_SHDW Register 1046
6-127. LPDDR2_NVM_TIM Register 1047

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6-128. LPDDR2_NVM_TIM_SHDW Register 1048
6-129. PWR_MGMT_CTRL Register 1049
6-130. PWR_MGMT_CTRL_SHDW Register 1050
6-131. LPDDR2_MODE_REG_DATA Register 1051
6-132. SDRAM_CONFIG Register 1052
6-133. LPDDR2_MODE_REG_CFG Register 1054
6-134. OCP_CONFIG Register 1055
6-135. OCP_CFG_VAL_1 Register 1056
6-136. OCP_CFG_VAL_2 Register 1057
6-137. IODFT_TLGC Register 1058
6-138. ELM Revision Register (ELM_REVISION) 1071
6-139. ELM System Configuration Register (ELM_SYSCONFIG) 1071
6-140. ELM System Status Register (ELM_SYSSTATUS) 1072
6-141. ELM Interrupt Status Register (ELM_IRQSTATUS) 1073
6-142. ELM Interrupt Enable Register (ELM_IRQENABLE) 1075
6-143. ELM Location Configuration Register (ELM_LOCATION_CONFIG) 1076
6-144. ELM Page Definition Register (ELM_PAGE_CTRL) 1077
6-145. ELM_SYNDROME_FRAGMENT_0_i Register 1078
6-146. ELM_SYNDROME_FRAGMENT_1_i Register 1078
6-147. ELM_SYNDROME_FRAGMENT_2_i Register 1078
6-148. ELM_SYNDROME_FRAGMENT_3_i Register 1079
6-149. ELM_SYNDROME_FRAGMENT_4_i Register 1079
6-150. ELM_SYNDROME_FRAGMENT_5_i Register 1079
6-151. ELM_SYNDROME_FRAGMENT_6_i Register 1080
6-152. ELM_LOCATION_STATUS_i Register 1080
6-153. ELM_ERROR_LOCATION_0-15_i Registers 1081
7-1. PM_CEFUSE_PWRSTCTRL Register 1086
7-2. PM_CEFUSE_PWRSTST Register 1087
7-3. PRM_RSTCTRL Register 1089
7-4. PRM_SRAM_COUNT Register 1090
7-5. PRM_LDO_SRAM_CORE_SETUP Register 1091
7-6. PRM_LDO_SRAM_CORE_CTRL Register 1092
7-7. PRM_LDO_SRAM_MPU_SETUP Register 1093
7-8. PRM_LDO_SRAM_MPU_CTRL Register 1094
7-9. PRM_RSTTIME Register 1095
7-10. PRM_RSTST Register 1096
7-11. PM_MPU_PWRSTCTRL Register 1098
7-12. PM_MPU_PWRSTST Register 1099
7-13. RM_MPU_RSTST Register 1100
7-14. RM_PER_RSTCTRL Register 1102
7-15. PM_PER_PWRSTCTRL Register 1103
7-16. RM_PER_RSTST Register 1104
7-17. PM_PER_PWRSTST Register 1105
7-18. PM_RTC_PWRSTCTRL Register 1107
7-19. PM_RTC_PWRSTST Register 1108
7-20. RM_WKUP_RSTCTRL Register 1110
7-21. RM_WKUP_RSTST Register 1111
7-22. PM_WKUP_PWRSTCTRL Register 1112
7-23. PM_WKUP_PWRSTST Register 1113

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7-24. CM_CEFUSE_CLKSTCTRL Register 1115
7-25. CM_CEFUSE_CEFUSE_CLKCTRL Register 1116
7-26. CM_CLKOUT_CTRL Register 1118
7-27. CLKSEL_TIMER7_CLK Register 1120
7-28. CLKSEL_TIMER4_CLK Register 1121
7-29. CM_MAC_CLKSEL Register 1122
7-30. CLKSEL_TIMER5_CLK Register 1123
7-31. CLKSEL_TIMER6_CLK Register 1124
7-32. CM_CPTS_RFT_CLKSEL Register 1125
7-33. CLKSEL_TIMER1MS_CLK Register 1126
7-34. CLKSEL_TIMER2_CLK Register 1127
7-35. CLKSEL_GFX_FCLK Register 1128
7-36. CLKSEL_ICSS_OCP_CLK Register 1129
7-37. CLKSEL_LCDC_PIXEL_CLK Register 1130
7-38. CLKSEL_WDT1_CLK Register 1131
7-39. CLKSEL_GPIO0_DBCLK Register 1132
7-40. CLKSEL_TIMER3_CLK Register 1133
7-41. CM_GFX_L3_CLKSTCTRL Register 1134
7-42. CM_GFX_L4LS_GFX_CLKSTCTRL Register 1135
7-43. CM_GFX_MMUCFG_CLKCTRL Register 1136
7-44. CM_GFX_MMUDATA_CLKCTRL Register 1137
7-45. CM_GFX_GFX_CLKCTRL Register 1138
7-46. CM_GFX_BITBLT_CLKCTRL Register 1139
7-47. CM_MPU_CLKSTCTRL Register 1140
7-48. CM_MPU_MPU_CLKCTRL Register 1141
7-49. CM_PER_L4LS_CLKSTCTRL Register 1146
7-50. CM_PER_L4FW_CLKCTRL Register 1148
7-51. CM_PER_MCASP1_CLKCTRL Register 1149
7-52. CM_PER_UART1_CLKCTRL Register 1150
7-53. CM_PER_UART2_CLKCTRL Register 1151
7-54. CM_PER_UART3_CLKCTRL Register 1152
7-55. CM_PER_L3_CLKSTCTRL Register 1153
7-56. CM_PER_UART4_CLKCTRL Register 1154
7-57. CM_PER_TIMER7_CLKCTRL Register 1155
7-58. CM_PER_TIMER2_CLKCTRL Register 1156
7-59. CM_PER_TIMER3_CLKCTRL Register 1157
7-60. CM_PER_TIMER4_CLKCTRL Register 1158
7-61. CM_PER_MCASP2_CLKCTRL Register 1159
7-62. CM_PER_RNG_CLKCTRL Register 1160
7-63. CM_PER_AES0_CLKCTRL Register 1161
7-64. CM_PER_AES1_CLKCTRL Register 1162
7-65. CM_PER_DES_CLKCTRL Register 1163
7-66. Reserved Register 1164
7-67. CM_PER_SHA0_CLKCTRL Register 1165
7-68. CM_PER_PKA_CLKCTRL Register 1166
7-69. CM_PER_GPIO6_CLKCTRL Register 1167
7-70. CM_PER_GPIO1_CLKCTRL Register 1168
7-71. CM_PER_GPIO2_CLKCTRL Register 1169
7-72. CM_PER_GPIO3_CLKCTRL Register 1170

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7-73. CM_PER_GPIO4_CLKCTRL Register 1171
7-74. CM_PER_TPCC_CLKCTRL Register 1172
7-75. CM_PER_DCAN0_CLKCTRL Register 1173
7-76. CM_PER_DCAN1_CLKCTRL Register 1174
7-77. CM_PER_CPGMAC0_CLKCTRL Register 1175
7-78. CM_PER_SPARE_CLKCTRL Register 1176
7-79. CM_PER_EPWMSS1_CLKCTRL Register 1177
7-80. CM_PER_EMIF_FW_CLKCTRL Register 1178
7-81. CM_PER_EPWMSS0_CLKCTRL Register 1179
7-82. CM_PER_EPWMSS2_CLKCTRL Register 1180
7-83. CM_PER_L3_INSTR_CLKCTRL Register 1181
7-84. CM_PER_L3_CLKCTRL Register 1182
7-85. CM_PER_IEEE5000_CLKCTRL Register 1183
7-86. CM_PER_ICSS_CLKCTRL Register 1184
7-87. CM_PER_TIMER5_CLKCTRL Register 1185
7-88. CM_PER_LCDC_CLKCTRL Register 1186
7-89. CM_PER_TIMER6_CLKCTRL Register 1187
7-90. CM_PER_MMC1_CLKCTRL Register 1188
7-91. CM_PER_MMC2_CLKCTRL Register 1189
7-92. CM_PER_TPTC1_CLKCTRL Register 1190
7-93. CM_PER_TPTC2_CLKCTRL Register 1191
7-94. CM_PER_GPIO5_CLKCTRL Register 1192
7-95. CM_PER_SPINLOCK_CLKCTRL Register 1193
7-96. CM_PER_MAILBOX0_CLKCTRL Register 1194
7-97. CM_PER_USB0_CLKCTRL Register 1195
7-98. CM_PER_L4HS_CLKSTCTRL Register 1196
7-99. CM_PER_L4HS_CLKCTRL Register 1197
7-100. CM_PER_MSTR_EXPS_CLKCTRL Register 1198
7-101. CM_PER_SLV_EXPS_CLKCTRL Register 1199
7-102. CM_PER_OCPWP_L3_CLKSTCTRL Register 1200
7-103. CM_PER_OCPWP_CLKCTRL Register 1201
7-104. CM_PER_MAILBOX1_CLKCTRL Register 1202
7-105. CM_PER_SPARE0_CLKCTRL Register 1203
7-106. CM_PER_SPARE1_CLKCTRL Register 1204
7-107. CM_PER_MLB_CLKCTRL Register 1205
7-108. CM_PER_ICSS_CLKSTCTRL Register 1206
7-109. CM_PER_CPSW_CLKSTCTRL Register 1207
7-110. CM_PER_LCDC_CLKSTCTRL Register 1208
7-111. CM_PER_CLKDIV32K_CLKCTRL Register 1209
7-112. CM_PER_CLK_24MHZ_CLKSTCTRL Register 1210
7-113. CM_PER_TPTC0_CLKCTRL Register 1211
7-114. CM_PER_L3S_CLKSTCTRL Register 1212
7-115. CM_PER_EMIF_CLKCTRL Register 1213
7-116. CM_PER_OCMCRAM_CLKCTRL Register 1214
7-117. CM_PER_GPMC_CLKCTRL Register 1215
7-118. CM_PER_MCASP0_CLKCTRL Register 1216
7-119. CM_PER_UART5_CLKCTRL Register 1217
7-120. CM_PER_MMC0_CLKCTRL Register 1218
7-121. CM_PER_ELM_CLKCTRL Register 1219

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7-122. CM_PER_I2C2_CLKCTRL Register 1220
7-123. CM_PER_I2C1_CLKCTRL Register 1221
7-124. CM_PER_SPI0_CLKCTRL Register 1222
7-125. CM_PER_L4FW_CLKSTCTRL Register 1223
7-126. CM_PER_SPI1_CLKCTRL Register 1224
7-127. CM_PER_SPI2_CLKCTRL Register 1225
7-128. CM_PER_SPI3_CLKCTRL Register 1226
7-129. CM_PER_L4LS_CLKCTRL Register 1227
7-130. CM_RTC_RTC_CLKCTRL Register 1228
7-131. CM_RTC_CLKSTCTRL Register 1229
7-132. CM_WKUP_CLKSTCTRL Register 1233
7-133. CM_SSC_MODFREQDIV_DPLL_CORE Register 1234
7-134. CM_CLKSEL_DPLL_CORE Register 1235
7-135. CM_AUTOIDLE_DPLL_PER Register 1236
7-136. CM_IDLEST_DPLL_PER Register 1237
7-137. CM_SSC_DELTAMSTEP_DPLL_PER Register 1238
7-138. CM_WKUP_L4WKUP_CLKCTRL Register 1239
7-139. CM_SSC_MODFREQDIV_DPLL_PER Register 1240
7-140. CM_CLKDCOLDO_DPLL_PER Register 1241
7-141. CM_DIV_M4_DPLL_CORE Register 1242
7-142. CM_DIV_M5_DPLL_CORE Register 1243
7-143. CM_CLKMODE_DPLL_MPU Register 1244
7-144. CM_CLKMODE_DPLL_PER Register 1246
7-145. CM_CLKMODE_DPLL_CORE Register 1247
7-146. CM_CLKMODE_DPLL_DDR Register 1249
7-147. CM_CLKMODE_DPLL_DISP Register 1251
7-148. CM_CLKSEL_DPLL_PERIPH Register 1253
7-149. CM_WKUP_TIMER0_CLKCTRL Register 1254
7-150. CM_DIV_M2_DPLL_DDR Register 1255
7-151. CM_DIV_M2_DPLL_DISP Register 1256
7-152. CM_DIV_M2_DPLL_MPU Register 1257
7-153. CM_DIV_M2_DPLL_PER Register 1258
7-154. CM_WKUP_WKUP_M3_CLKCTRL Register 1259
7-155. CM_WKUP_UART0_CLKCTRL Register 1260
7-156. CM_WKUP_I2C0_CLKCTRL Register 1261
7-157. CM_WKUP_ADC_TSC_CLKCTRL Register 1262
7-158. CM_WKUP_SMARTREFLEX0_CLKCTRL Register 1263
7-159. CM_WKUP_TIMER1_CLKCTRL Register 1264
7-160. CM_WKUP_DEBUGSS_CLKCTRL Register 1265
7-161. CM_WKUP_SMARTREFLEX1_CLKCTRL Register 1266
7-162. CM_L4_WKUP_AON_CLKSTCTRL Register 1267
7-163. CM_WKUP_WDT0_CLKCTRL Register 1268
7-164. CM_WKUP_WDT1_CLKCTRL Register 1269
7-165. CM_DIV_M6_DPLL_CORE Register 1270
7-166. CM_L3_AON_CLKSTCTRL Register 1271
7-167. CM_AUTOIDLE_DPLL_MPU Register 1272
7-168. CM_IDLEST_DPLL_MPU Register 1273
7-169. CM_SSC_DELTAMSTEP_DPLL_MPU Register 1274
7-170. CM_WKUP_CONTROL_CLKCTRL Register 1275

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7-171. CM_SSC_MODFREQDIV_DPLL_MPU Register 1276
7-172. CM_CLKSEL_DPLL_MPU Register 1277
7-173. CM_AUTOIDLE_DPLL_DDR Register 1278
7-174. CM_IDLEST_DPLL_DDR Register 1279
7-175. CM_SSC_DELTAMSTEP_DPLL_DDR Register 1280
7-176. CM_SSC_MODFREQDIV_DPLL_DDR Register 1281
7-177. CM_CLKSEL_DPLL_DDR Register 1282
7-178. CM_AUTOIDLE_DPLL_DISP Register 1283
7-179. CM_IDLEST_DPLL_DISP Register 1284
7-180. CM_SSC_DELTAMSTEP_DPLL_DISP Register 1285
7-181. CM_WKUP_GPIO0_CLKCTRL Register 1286
7-182. CM_SSC_MODFREQDIV_DPLL_DISP Register 1287
7-183. CM_CLKSEL_DPLL_DISP Register 1288
7-184. CM_AUTOIDLE_DPLL_CORE Register 1289
7-185. CM_IDLEST_DPLL_CORE Register 1290
7-186. CM_SSC_DELTAMSTEP_DPLL_CORE Register 1291
9-1. Agent Control Register - IA Block (L3_IA_AGENT_CONTROL) 1300
9-2. Agent Status Register (L3_IA_AGENT_STATUS) 1302
9-3. Error Log Register - IA Block (L3_IA_ERROR_LOG) 1304
9-4. Error Log Address Register - IA Block (L3_IA_ERROR_LOG_ADDR) 1305
9-5. Agent Control Register - TA Block (L3_TA_AGENT_CONTROL) 1308
9-6. Agent Status Register (L3_TA_AGENT_STATUS) 1309
9-7. Error Log Register - TA Block (L3_TA_ERROR_LOG) 1310
9-8. Error Log Address Register - TA Block (L3_TA_ERROR_LOG_ADDR) 1311
9-9. L3_RT_NETWORK 1313
9-10. L3_RT_INITID_READBACK 1313
9-11. L3_RT_NETWORK_CONTROL 1314
9-12. L3_PM_ERROR_LOG 1317
9-13. L3_PM_CONTROL 1318
9-14. L3_PM_ERROR_CLEAR_SINGLE 1319
9-15. L3_PM_ERROR_CLEAR_MULTI 1319
9-16. L3_PM_REQ_INFO_PERMISSION_i 1320
9-17. L3_PM_READ_PERMISSION_i 1321
9-18. L3_PM_WRITE_PERMISSION_i 1322
9-19. L3_PM_ADDR_MATCH_k 1324
9-20. L3_SI_CONTROL 1327
9-21. L3_SI_FLAG_STATUS_0 1328
9-22. L3_SI_FLAG_STATUS_1 1328
9-23. L4_IA_AGENT_CONTROL_L 1333
9-24. L4_IA_AGENT_STATUS_L 1334
9-25. L4_IA_ERROR_LOG_L 1334
9-26. L4_TA_AGENT_CONTROL_L 1342
9-27. L4_TA_AGENT_CONTROL_H 1343
9-28. L4_TA_AGENT_STATUS_L 1343
9-29. L4_LA_NETWORK_H 1345
9-30. L4_LA_INITIATOR_INFO_L 1345
9-31. L4_LA_INITIATOR_INFO_H 1346
9-32. L4_LA_NETWORK_CONTROL_L 1347
9-33. L4_LA_NETWORK_CONTROL_H 1348

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9-34. L4_AP_SEGMENT_i_L 1355
9-35. L4_AP_SEGMENT_i_H 1356
9-36. L4_AP_PROT_GROUP_MEMBERS_k_L 1356
9-37. L4_AP_PROT_GROUP_ROLES_k_L 1357
9-38. L4_AP_REGION_l_L 1357
9-39. L4_AP_REGION_l_H 1358
10-1. Peripheral ID Register (PID) 1364
10-2. EDMA3CC Configuration Register (CCCFG) 1365
10-3. DMA Channel Map *n* Registers (DCHMAP*n*) 1367
10-4. QDMA Channel Map *n* Registers (QCHMAP*n*) 1368
10-5. DMA Channel Queue *n* Number Registers (DMAQNUM*n*) 1369
10-6. QDMA Channel Queue Number Register (QDMAQNUM) 1370
10-7. Queue Priority Register (QUEPRI) 1371
10-8. Event Missed Register (EMR) 1372
10-9. Event Missed Register High (EMRH) 1372
10-10. Event Missed Clear Register (EMCR) 1373
10-11. Event Missed Clear Register High (EMCRH) 1373
10-12. QDMA Event Missed Register (QEMR) 1374
10-13. QDMA Event Missed Clear Register (QEMCR) 1375
10-14. EDMA3CC Error Register (CCERR) 1375
10-15. EDMA3CC Error Clear Register (CCERRCLR) 1376
10-16. Error Evaluation Register (EEVAL) 1378
10-17. DMA Region Access Enable Register for Region *m* (DRAE*m*) 1379
10-18. DMA Region Access Enable High Register for Region *m* (DRAEH*m*) 1379
10-19. QDMA Region Access Enable for Region *m* (QRAE*m*)32-bit, 2 Rows 1380
10-20. Event Queue Entry Registers (Q*x*E*y*) 1381
10-21. Queue Status Register *n* (QSTAT*n*) 1382
10-22. Queue Watermark Threshold A Register (QWMTHRA) 1383
10-23. EDMA3CC Status Register (CCSTAT) 1384
10-24. Memory Protection Fault Address Register (MPFAR) 1386
10-25. Memory Protection Fault Status Register (MPFSR) 1387
10-26. Memory Protection Fault Command Register (MPFCR) 1388
10-27. Memory Protection Page Attribute Register (MPPA*n*) 1389
10-28. Event Register (ER) 1391
10-29. Event Register High (ERH) 1391
10-30. Event Clear Register (ECR) 1392
10-31. Event Clear Register High (ECRH) 1392
10-32. Event Set Register (ESR) 1393
10-33. Event Set Register High (ESRH) 1394
10-34. Chained Event Register (CER) 1395
10-35. Chained Event Register High (CERH) 1395
10-36. Event Enable Register (EER) 1397
10-37. Event Enable Register High (EERH) 1397
10-38. Event Enable Clear Register (EECR) 1398
10-39. Event Enable Clear Register High (EECRH) 1398
10-40. Event Enable Set Register (EESR) 1399
10-41. Event Enable Set Register High (EESRH) 1399
10-42. Secondary Event Register (SER) 1400
10-43. Secondary Event Register High (SERH) 1400

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10-44. Secondary Event Clear Register (SECR) 1401
10-45. Secondary Event Clear Register High (SECRH) 1401
10-46. Interrupt Enable Register (IER) 1402
10-47. Interrupt Enable Register High (IERH) 1402
10-48. Interrupt Enable Clear Register (IECR) 1403
10-49. Interrupt Enable Clear Register High (IECRH) 1403
10-50. Interrupt Enable Set Register (IESR) 1404
10-51. Interrupt Enable Set Register High (IESRH) 1404
10-52. Interrupt Pending Register (IPR) 1405
10-53. Interrupt Pending Register High (IPRH) 1405
10-54. Interrupt Clear Register (ICR) 1406
10-55. Interrupt Clear Register High (ICRH) 1406
10-56. Interrupt Evaluate Register (IEVAL) 1407
10-57. QDMA Event Register (QER) 1408
10-58. QDMA Event Enable Register (QEER) 1409
10-59. QDMA Event Enable Clear Register (QEECR) 1410
10-60. QDMA Event Enable Set Register (QEESR) 1411
10-61. QDMA Secondary Event Register (QSER) 1412
10-62. QDMA Secondary Event Clear Register (QSECR) 1413
10-63. Peripheral ID Register (PID) 1415
10-64. EDMA3TC Configuration Register (TCCFG) 1416
10-65. EDMA3TC Channel Status Register (TCSTAT) 1417
10-66. Error Register (ERRSTAT) 1419
10-67. Error Enable Register (ERREN) 1420
10-68. Error Clear Register (ERRCLR) 1421
10-69. Error Details Register (ERRDET) 1422
10-70. Error Interrupt Command Register (ERRCMD) 1423
10-71. Read Rate Register (RDRATE) 1424
10-72. Source Active Options Register (SAOPT) 1425
10-73. Source Active Source Address Register (SASRC) 1427
10-74. Source Active Count Register (SACNT) 1427
10-75. Source Active Destination Address Register (SADST) 1428
10-76. Source Active Source B-Dimension Index Register (SABIDX) 1428
10-77. Source Active Memory Protection Proxy Register (SAMPPRXY) 1429
10-78. Source Active Count Reload Register (SACNTRLD) 1430
10-79. Source Active Source Address B-Reference Register (SASRCBREF) 1430
10-80. Source Active Destination Address B-Reference Register (SADSTBREF) 1431
10-81. Destination FIFO Options Register (DFOPT*n*) 1432
10-82. Destination FIFO Source Address Register (DFSRC*n*) 1434
10-83. Destination FIFO Count Register (DFCNT*n*) 1434
10-84. Destination FIFO Destination Address Register (DFDST*n*) 1435
10-85. Destination FIFO B-Index Register (DFBIDX*n*) 1435
10-86. Destination FIFO Memory Protection Proxy Register (DFMPPRXY*n*) 1436
10-87. Destination FIFO Count Reload Register (DFCNTRLD*n*) 1437
10-88. Destination FIFO Source Address B-Reference Register (DFSRCBREF*n*) 1437
10-89. Destination FIFO Destination Address B-Reference Register (DFDSTBREF*n*) 1438
11-1. TSC_ADC Integration 1441
11-2. Functional Block Diagram 1444
11-3. Sequencer FSM 1447

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11-4. Example Timing Diagram for Sequencer 1447
11-5. REVISION Register 1450
11-6. STEPCONFIG4 Register 1451
11-7. STEPDELAY4 Register 1452
11-8. STEPCONFIG5 Register 1453
11-9. STEPDELAY5 Register 1454
11-10. STEPCONFIG6 Register 1455
11-11. IRQSTATUS_RAW Register 1456
11-12. STEPDELAY6 Register 1458
11-13. STEPCONFIG7 Register 1459
11-14. STEPDELAY7 Register 1460
11-15. STEPCONFIG8 Register 1461
11-16. STEPDELAY8 Register 1462
11-17. STEPCONFIG9 Register 1463
11-18. STEPDELAY9 Register 1464
11-19. STEPCONFIG10 Register 1465
11-20. STEPDELAY10 Register 1466
11-21. STEPCONFIG11 Register 1467
11-22. IRQSTATUS Register 1468
11-23. STEPDELAY11 Register 1470
11-24. STEPCONFIG12 Register 1471
11-25. STEPDELAY12 Register 1472
11-26. STEPCONFIG13 Register 1473
11-27. STEPDELAY13 Register 1474
11-28. STEPCONFIG14 Register 1475
11-29. STEPDELAY14 Register 1476
11-30. STEPCONFIG15 Register 1477
11-31. STEPDELAY15 Register 1478
11-32. STEPCONFIG16 Register 1479
11-33. IRQENABLE_SET Register 1480
11-34. STEPDELAY16 Register 1482
11-35. FIFO0COUNT Register 1483
11-36. FIFO0THRESHOLD Register 1484
11-37. DMA0REQ Register 1485
11-38. FIFO1COUNT Register 1486
11-39. FIFO1THRESHOLD Register 1487
11-40. DMA1REQ Register 1488
11-41. FIFO0DATA Register 1489
11-42. FIFO1DATA Register 1490
11-43. IRQENABLE_CLR Register 1491
11-44. IRQWAKEUP Register 1493
11-45. DMAENABLE_SET Register 1494
11-46. DMAENABLE_CLR Register 1495
11-47. SYSCONFIG Register 1496
11-48. CTRL Register 1497
11-49. ADCSTAT Register 1498
11-50. ADCRANGE Register 1499
11-51. ADC_CLKDIV Register 1500
11-52. ADC_MISC Register 1501

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11-53. STEPENABLE Register 1502
11-54. IDLECONFIG Register 1503
11-55. TS_CHARGE_STEPCONFIG Register 1504
11-56. TS_CHARGE_DELAY Register 1505
11-57. STEPCONFIG1 Register 1506
11-58. IRQ_EOI Register 1507
11-59. STEPDELAY1 Register 1508
11-60. STEPCONFIG2 Register 1509
11-61. STEPDELAY2 Register 1510
11-62. STEPCONFIG3 Register 1511
11-63. STEPDELAY3 Register 1512
12-1. LCD Controller 1514
12-2. LCD Controller Integration 1516
12-3. Input and Output Clocks 1518
12-4. LIDD Mode HD44780 Write Timing Diagram 1524
12-5. LIDD Mode HD44780 Read Timing Diagram 1524
12-6. LIDD Mode 6800 Write Timing Diagram 1524
12-7. LIDD Mode 6800 Read Timing Diagram 1525
12-8. LIDD Mode 6800 Status Timing Diagram 1525
12-9. LIDD Mode 8080 Write Timing Diagram 1525
12-10. LIDD Mode 8080 Read Timing Diagram 1526
12-11. LIDD Mode 8080 Status Timing Diagram 1526
12-12. Logical Data Path for Raster Controller 1528
12-13. Frame Buffer Structure 1529
12-14. 16-Entry Palette/Buffer Format (1, 2, 4, 12, 16 BPP) 1530
12-15. 256-Entry Palette/Buffer Format (8 BPP) 1531
12-16. 16-BPP Data Memory Organization (TFT Mode Only)—Little Endian 1531
12-17. 12-BPP Data Memory Organization—Little Endian 1532
12-18. 8-BPP Data Memory Organization 1532
12-19. 4-BPP Data Memory Organization 1532
12-20. 2-BPP Data Memory Organization 1533
12-21. 1-BPP Data Memory Organization 1533
12-22. Monochrome and Color Output 1535
12-23. Raster Mode Display Format 1536
12-24. Raster Mode Passive (STN) Timing Diagram 1537
12-25. Raster Mode Active (TFT) Timing Diagram 1538
12-26. pid Register 1540
12-27. IRQENABLE_CLEAR Register 1541
12-28. IRQEOI_VECTOR Register 1543
12-29. CLKC_ENABLE Register 1544
12-30. CLKC_RESET Register 1545
12-31. LIDD_CTRL Register 1546
12-32. LIDD_CS0_CONF Register 1548
12-33. LIDD_CS0_ADDR Register 1549
12-34. reg29 Register 1550
12-35. reg30 Register 1551
12-36. LIDD_CS0_DATA Register 1552
12-37. reg31 Register 1553
12-38. LIDD_CS1_CONF Register 1554

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12-39. LIDD_CS1_ADDR Register 1555
12-40. LIDD_CS1_DATA Register 1556
12-41. CTRL Register 1557
12-42. RASTER_CTRL Register 1558
12-43. RASTER_TIMING_0 Register 1560
12-44. RASTER_TIMING_1 Register 1561
12-45. RASTER_TIMING_2 Register 1562
12-46. RASTER_SUBPANEL Register 1564
12-47. RASTER_SUBPANEL2 Register 1565
12-48. LCDDMA_CTRL Register 1566
12-49. LCDDMA_FB0_BASE Register 1567
12-50. LCDDMA_FB0_CEILING Register 1568
12-51. LCDDMA_FB1_BASE Register 1569
12-52. LCDDMA_FB1_CEILING Register 1570
12-53. SYSCONFIG Register 1571
12-54. IRQSTATUS_RAW Register 1572
12-55. IRQSTATUS Register 1573
12-56. IRQENABLE_SET Register 1574
13-1. Ethernet Switch Integration 1580
13-2. CPSW_3G Block Diagram 1583
13-3. The Network static with AVB 1593
13-4. IEEE 1722 Packets 1594
13-5. GMII Interface Connections 1597
13-6. RMII Interface Connections 1599
13-7. RGMII Interface Connections 1600
13-8. CPTS Block Diagram 1607
13-9. CPTS RCLK Source 1607
13-10. Event FIFO Misalignment Condition 1608
13-11. HW1/4_TSP_PUSH Connection 1608
13-12. Partial Ethernet-II frames showing register mapping of EtherTypes for a simple frame (1), a single 1Q tag added (2), and two 1Q tags added (3). 1609
13-13. 1626
13-14. 1626
13-15. IDVER Register 1630
13-16. PRESCALE Register 1631
13-17. UNKNOWN_VLAN Register 1632
13-18. TBLCTL Register 1633
13-19. TBLW2 Register 1634
13-20. TBLW1 Register 1635
13-21. TBLW0 Register 1636
13-22. PORTCTL0 Register 1637
13-23. PORTCTL1 Register 1638
13-24. PORTCTL2 Register 1639
13-25. PORTCTL3 Register 1640
13-26. CONTROL Register 1641
13-27. PORTCTL4 Register 1643
13-28. PORTCTL5 Register 1644
13-29. TX_IDVER Register 1648
13-30. TX_INTSTAT_RAW Register 1649

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13-31. TX_INTSTAT_MASKED Register 1650
13-32. TX_INTMASK_SET Register 1651
13-33. TX_INTMASK_CLEAR Register 1652
13-34. CPDMA_IN_VECTOR Register 1653
13-35. CPDMA_EOI_VECTOR Register 1654
13-36. RX_IDVER Register 1655
13-37. RX_INTSTAT_RAW Register 1656
13-38. RX_INTSTAT_MASKED Register 1657
13-39. RX_INTMASK_SET Register 1658
13-40. RX_INTMASK_CLEAR Register 1659
13-41. DMA_INTSTAT_RAW Register 1660
13-42. DMA_INTSTAT_MASKED Register 1661
13-43. DMA_INTMASK_SET Register 1662
13-44. DMA_INTMASK_CLEAR Register 1663
13-45. RX0_PENDTHRESH Register 1664
13-46. RX1_PENDTHRESH Register 1665
13-47. RX_CONTROL Register 1666
13-48. RX2_PENDTHRESH Register 1667
13-49. RX3_PENDTHRESH Register 1668
13-50. RX4_PENDTHRESH Register 1669
13-51. RX5_PENDTHRESH Register 1670
13-52. RX6_PENDTHRESH Register 1671
13-53. RX7_PENDTHRESH Register 1672
13-54. RX0_FREEBUFFER Register 1673
13-55. RX1_FREEBUFFER Register 1674
13-56. RX2_FREEBUFFER Register 1675
13-57. RX3_FREEBUFFER Register 1676
13-58. RX_TEARDOWN Register 1677
13-59. RX4_FREEBUFFER Register 1678
13-60. RX5_FREEBUFFER Register 1679
13-61. RX6_FREEBUFFER Register 1680
13-62. RX7_FREEBUFFER Register 1681
13-63. TX0_HDP Register 1682
13-64. TX1_HDP Register 1683
13-65. TX2_HDP Register 1684
13-66. TX3_HDP Register 1685
13-67. TX4_HDP Register 1686
13-68. TX5_HDP Register 1687
13-69. TX6_HDP Register 1688
13-70. TX7_HDP Register 1689
13-71. RX0_HDP Register 1690
13-72. RX1_HDP Register 1691
13-73. RX2_HDP Register 1692
13-74. RX3_HDP Register 1693
13-75. RX4_HDP Register 1694
13-76. RX5_HDP Register 1695
13-77. RX6_HDP Register 1696
13-78. RX7_HDP Register 1697
13-79. TX0_CP Register 1698

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13-80. TX1_CP Register 1699
13-81. TX2_CP Register 1700
13-82. TX3_CP Register 1701
13-83. TX4_CP Register 1702
13-84. TX5_CP Register 1703
13-85. TX6_CP Register 1704
13-86. TX7_CP Register 1705
13-87. RX0_CP Register 1706
13-88. RX1_CP Register 1707
13-89. RX2_CP Register 1708
13-90. RX3_CP Register 1709
13-91. RX4_CP Register 1710
13-92. RX5_CP Register 1711
13-93. RX6_CP Register 1712
13-94. RX7_CP Register 1713
13-95. CPDMA_SOFT_RESET Register 1714
13-96. DMACONTROL Register 1715
13-97. DMASTATUS Register 1717
13-98. TX_CONTROL Register 1718
13-99. RX_BUFFER_OFFSET Register 1719
13-100. EMCONTROL Register 1720
13-101. TX_PRI0_RATE Register 1721
13-102. TX_PRI1_RATE Register 1722
13-103. TX_PRI2_RATE Register 1723
13-104. TX_PRI3_RATE Register 1724
13-105. TX_PRI4_RATE Register 1725
13-106. TX_PRI5_RATE Register 1726
13-107. TX_PRI6_RATE Register 1727
13-108. TX_PRI7_RATE Register 1728
13-109. TX_TEARDOWN Register 1729
13-110. CPTS_IDVER Register 1731
13-111. CPTS_TS_PUSH Register 1732
13-112. CPTS_TS_LOAD_VAL Register 1733
13-113. CPTS_TS_LOAD_EN Register 1734
13-114. CPTS_INTSTAT_RAW Register 1735
13-115. CPTS_INTSTAT_MASKED Register 1736
13-116. CPTS_CONTROL Register 1737
13-117. CPTS_INT_ENABLE Register 1738
13-118. CPTS_EVENT_POP Register 1739
13-119. CPTS_EVENT_LOW Register 1740
13-120. CPTS_EVENT_HIGH Register 1741
13-121. CPTS_RFTCLK_SEL Register 1742
13-122. P0_CONTROL Register 1745
13-123. P0_BLK_CNT Register 1746
13-124. P0_TX_IN_CTL Register 1747
13-125. P0_PORT_VLAN Register 1748
13-126. P0_TX_PRI_MAP Register 1749
13-127. P1_CONTROL Register 1750
13-128. P1_MAX_BLKS Register 1752

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13-129. P1_BLK_CNT Register 1753
13-130. P1_TX_IN_CTL Register 1754
13-131. P1_PORT_VLAN Register 1755
13-132. P0_CPDMA_TX_PRI_MAP Register 1756
13-133. P1_TX_PRI_MAP Register 1757
13-134. P1_TS_SEQ_MTYPE Register 1758
13-135. P1_SA_LO Register 1759
13-136. P1_SA_HI Register 1760
13-137. P1_SEND_PERCENT Register 1761
13-138. P1_RX_DSCP_PRI_MAP0 Register 1762
13-139. P1_RX_DSCP_PRI_MAP1 Register 1763
13-140. P1_RX_DSCP_PRI_MAP2 Register 1764
13-141. P1_RX_DSCP_PRI_MAP3 Register 1765
13-142. P0_CPDMA_RX_CH_MAP Register 1766
13-143. P1_RX_DSCP_PRI_MAP4 Register 1767
13-144. P1_RX_DSCP_PRI_MAP5 Register 1768
13-145. P1_RX_DSCP_PRI_MAP6 Register 1769
13-146. P1_RX_DSCP_PRI_MAP7 Register 1770
13-147. P0_RX_DSCP_PRI_MAP0 Register 1771
13-148. P2_CONTROL Register 1772
13-149. P0_RX_DSCP_PRI_MAP1 Register 1774
13-150. P2_MAX_BLKS Register 1775
13-151. P2_BLK_CNT Register 1776
13-152. P2_TX_IN_CTL Register 1777
13-153. P2_PORT_VLAN Register 1778
13-154. P2_TX_PRI_MAP Register 1779
13-155. P2_TS_SEQ_MTYPE Register 1780
13-156. P2_SA_LO Register 1781
13-157. P2_SA_HI Register 1782
13-158. P2_SEND_PERCENT Register 1783
13-159. P0_RX_DSCP_PRI_MAP2 Register 1784
13-160. P2_RX_DSCP_PRI_MAP0 Register 1785
13-161. P2_RX_DSCP_PRI_MAP1 Register 1786
13-162. P2_RX_DSCP_PRI_MAP2 Register 1787
13-163. P2_RX_DSCP_PRI_MAP3 Register 1788
13-164. P2_RX_DSCP_PRI_MAP4 Register 1789
13-165. P2_RX_DSCP_PRI_MAP5 Register 1790
13-166. P2_RX_DSCP_PRI_MAP6 Register 1791
13-167. P2_RX_DSCP_PRI_MAP7 Register 1792
13-168. P0_RX_DSCP_PRI_MAP3 Register 1793
13-169. P0_RX_DSCP_PRI_MAP4 Register 1794
13-170. P0_RX_DSCP_PRI_MAP5 Register 1795
13-171. P0_RX_DSCP_PRI_MAP6 Register 1796
13-172. P0_RX_DSCP_PRI_MAP7 Register 1797
13-173. P0_MAX_BLKS Register 1798
13-174. IDVER Register 1800
13-175. SOFT_RESET Register 1801
13-176. RX_MAXLEN Register 1802
13-177. BOFFTEST Register 1803

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13-178. RX_PAUSE Register 1804
13-179. TX_PAUSE Register 1805
13-180. EMCONTROL Register 1806
13-181. RX_PRI_MAP Register 1807
13-182. MACCONTROL Register 1808
13-183. TX_GAP Register 1810
13-184. MACSTATUS Register 1811
13-185. ID_VER Register 1813
13-186. STAT_PORT_EN Register 1814
13-187. PTYPE Register 1815
13-188. SOFT_IDLE Register 1816
13-189. THRU_RATE Register 1817
13-190. GAP_THRESH Register 1818
13-191. TX_START_WDS Register 1819
13-192. FLOW_CONTROL Register 1820
13-193. CONTROL Register 1821
13-194. VLAN_LTYPE Register 1822
13-195. TS_LTYPE Register 1823
13-196. DLR_LTYPE Register 1824
13-197. SOFT_RESET Register 1825
13-198. IDVER Register 1827
13-199. C2_RX_STAT Register 1828
13-200. C2_TX_STAT Register 1829
13-201. C2_MISC_STAT Register 1830
13-202. C0_RX_IMAX Register 1831
13-203. C0_TX_IMAX Register 1832
13-204. INT_CONTROL Register 1833
13-205. C1_RX_IMAX Register 1834
13-206. C1_TX_IMAX Register 1835
13-207. C2_RX_IMAX Register 1836
13-208. C2_TX_IMAX Register 1837
13-209. RGMII_CTL Register 1838
13-210. C0_RX_THRESH_EN Register 1839
13-211. C0_RX_EN Register 1840
13-212. C0_TX_EN Register 1841
13-213. C0_MISC_EN Register 1842
13-214. C1_RX_THRESH_EN Register 1843
13-215. C1_RX_EN Register 1844
13-216. SOFT_RESET Register 1845
13-217. C1_TX_EN Register 1846
13-218. C1_MISC_EN Register 1847
13-219. C2_RX_THRESH_EN Register 1848
13-220. C2_RX_EN Register 1849
13-221. C2_TX_EN Register 1850
13-222. C2_MISC_EN Register 1851
13-223. C0_RX_THRESH_STAT Register 1852
13-224. C0_RX_STAT Register 1853
13-225. C0_TX_STAT Register 1854
13-226. C0_MISC_STAT Register 1855

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13-227. CONTROL Register 1856
13-228. C1_RX_THRESH_STAT Register 1857
13-229. C1_RX_STAT Register 1858
13-230. C1_TX_STAT Register 1859
13-231. C1_MISC_STAT Register 1860
13-232. C2_RX_THRESH_STAT Register 1861
14-1. PWMSS Integration 1866
14-2. Multiple ePWM Modules 1869
14-3. Submodules and Signal Connections for an ePWM Module 1870
14-4. ePWM Submodules and Critical Internal Signal Interconnects 1871
14-5. Time-Base Submodule Block Diagram 1877
14-6. Time-Base Submodule Signals and Registers 1879
14-7. Time-Base Frequency and Period 1881
14-8. Time-Base Counter Synchronization Scheme 1 1882
14-9. Time-Base Up-Count Mode Waveforms 1884
14-10. Time-Base Down-Count Mode Waveforms 1885
14-11. Time-Base Up-Down-Count Waveforms, TBCTL[PHSDIR = 0] Count Down on Synchronization Event 1885
14-12. Time-Base Up-Down Count Waveforms, TBCTL[PHSDIR = 1] Count Up on Synchronization Event 1886
14-13. Counter-Compare Submodule 1887
14-14. Counter-Compare Submodule Signals and Registers 1887
14-15. Counter-Compare Event Waveforms in Up-Count Mode 1890
14-16. Counter-Compare Events in Down-Count Mode 1890
14-17. Counter-Compare Events in Up-Down-Count Mode, TBCTL[PHSDIR = 0] Count Down on Synchronization Event 1891
14-18. Counter-Compare Events in Up-Down-Count Mode, TBCTL[PHSDIR = 1] Count Up on Synchronization Event 1891
14-19. Action-Qualifier Submodule 1892
14-20. Action-Qualifier Submodule Inputs and Outputs 1893
14-21. Possible Action-Qualifier Actions for EPWMxA and EPWMxB Outputs 1894
14-22. Up-Down-Count Mode Symmetrical Waveform 1897
14-23. Up, Single Edge Asymmetric Waveform, With Independent Modulation on EPWMxA and EPWMxB—Active High 1898
14-24. Up, Single Edge Asymmetric Waveform With Independent Modulation on EPWMxA and EPWMxB—Active Low 1900
14-25. Up-Count, Pulse Placement Asymmetric Waveform With Independent Modulation on EPWMxA 1902
14-26. Up-Down-Count, Dual Edge Symmetric Waveform, With Independent Modulation on EPWMxA and EPWMxB — Active Low 1904
14-27. Up-Down-Count, Dual Edge Symmetric Waveform, With Independent Modulation on EPWMxA and EPWMxB — Complementary 1906
14-28. Up-Down-Count, Dual Edge Asymmetric Waveform, With Independent Modulation on EPWMxA—Active Low 1908
14-29. Dead-Band Generator Submodule 1910
14-30. Configuration Options for the Dead-Band Generator Submodule 1911
14-31. Dead-Band Waveforms for Typical Cases (0% < Duty < 100%) 1913
14-32. PWM-Chopper Submodule 1914
14-33. PWM-Chopper Submodule Signals and Registers 1915
14-34. Simple PWM-Chopper Submodule Waveforms Showing Chopping Action Only 1916
14-35. PWM-Chopper Submodule Waveforms Showing the First Pulse and Subsequent Sustaining Pulses 1916
14-36. PWM-Chopper Submodule Waveforms Showing the Pulse Width (Duty Cycle) Control of Sustaining Pulses 1917
14-37. Trip-Zone Submodule 1918

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14-38. Trip-Zone Submodule Mode Control Logic 1921
14-39. Trip-Zone Submodule Interrupt Logic 1921
14-40. Event-Trigger Submodule 1922
14-41. Event-Trigger Submodule Inter-Connectivity to Interrupt Controller 1923
14-42. Event-Trigger Submodule Showing Event Inputs and Prescaled Outputs 1923
14-43. Event-Trigger Interrupt Generator 1925
14-44. HRPWM System Interface 1926
14-45. Resolution Calculations for Conventionally Generated PWM 1927
14-46. Operating Logic Using MEP 1928
14-47. Required PWM Waveform for a Requested Duty = 40.5% 1930
14-48. Low % Duty Cycle Range Limitation Example When PWM Frequency = 1 MHz 1932
14-49. High % Duty Cycle Range Limitation Example when PWM Frequency = 1 MHz 1932
14-50. Simplified ePWM Module 1944
14-51. EPWM1 Configured as a Typical Master, EPWM2 Configured as a Slave 1945
14-52. Control of Four Buck Stages. Here $F_{PWM1} \neq F_{PWM2} \neq F_{PWM3} \neq F_{PWM4}$ 1946
14-53. Buck Waveforms for (Note: Only three bucks shown here) 1947
14-54. Control of Four Buck Stages. (Note: $F_{PWM2} = N \times F_{PWM1}$) 1949
14-55. Buck Waveforms for (Note: $F_{PWM2} = F_{PWM1}$) 1950
14-56. Control of Two Half-H Bridge Stages ($F_{PWM2} = N \times F_{PWM1}$) 1952
14-57. Half-H Bridge Waveforms for (Note: Here $F_{PWM2} = F_{PWM1}$) 1953
14-58. Control of Dual 3-Phase Inverter Stages as Is Commonly Used in Motor Control 1955
14-59. 3-Phase Inverter Waveforms for (Only One Inverter Shown) 1956
14-60. Configuring Two PWM Modules for Phase Control 1959
14-61. Timing Waveforms Associated With Phase Control Between 2 Modules 1960
14-62. Control of a 3-Phase Interleaved DC/DC Converter 1961
14-63. 3-Phase Interleaved DC/DC Converter Waveforms for 1962
14-64. Controlling a Full-H Bridge Stage ($F_{PWM2} = F_{PWM1}$) 1965
14-65. ZVS Full-H Bridge Waveforms 1966
14-66. Multiple eCAP Modules 1970
14-67. Capture and APWM Modes of Operation 1971
14-68. Capture Function Diagram 1972
14-69. Event Prescale Control 1973
14-70. Prescale Function Waveforms 1973
14-71. Continuous/One-shot Block Diagram 1974
14-72. Counter and Synchronization Block Diagram 1975
14-73. Interrupts in eCAP Module 1977
14-74. PWM Waveform Details Of APWM Mode Operation 1978
14-75. Capture Sequence for Absolute Time-Stamp, Rising Edge Detect 1981
14-76. Capture Sequence for Absolute Time-Stamp, Rising and Falling Edge Detect 1983
14-77. Capture Sequence for Delta Mode Time-Stamp, Rising Edge Detect 1985
14-78. Capture Sequence for Delta Mode Time-Stamp, Rising and Falling Edge Detect 1987
14-79. PWM Waveform Details of APWM Mode Operation 1989
14-80. Multichannel PWM Example Using 4 eCAP Modules 1991
14-81. Multiphase (channel) Interleaved PWM Example Using 3 eCAP Modules 1994
14-82. Time-Stamp Counter Register (TSCTR) 1996
14-83. Counter Phase Control Register (CTRPHS) 1997
14-84. Capture 1 Register (CAP1) 1997
14-85. Capture 2 Register (CAP2) 1998
14-86. Capture 3 Register (CAP3) 1998

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14-87. Capture 4 Register (CAP4) 1999
14-88. ECAP Control Register 1 (ECCTL1) 1999
14-89. ECAP Control Register 2 (ECCTL2) 2001
14-90. ECAP Interrupt Enable Register (ECEINT) 2003
14-91. ECAP Interrupt Flag Register (ECFLG) 2004
14-92. ECAP Interrupt Clear Register (ECCLR) 2005
14-93. ECAP Interrupt Forcing Register (ECFRC) 2006
14-94. Revision ID Register (REVID) 2007
14-95. Optical Encoder Disk 2008
14-96. QEP Encoder Output Signal for Forward/Reverse Movement 2009
14-97. Index Pulse Example 2009
14-98. Functional Block Diagram of the eQEP Peripheral 2012
14-99. Functional Block Diagram of Decoder Unit 2013
14-100. Quadrature Decoder State Machine 2015
14-101. Quadrature-clock and Direction Decoding 2015
14-102. Position Counter Reset by Index Pulse for 1000 Line Encoder (QPOSMAX = 3999 or F9Fh) 2017
14-103. Position Counter Underflow/Overflow (QPOSMAX = 4) 2018
14-104. Software Index Marker for 1000-line Encoder (QEPCTL[IEL] = 1) 2020
14-105. Strobe Event Latch (QEPCTL[SEL] = 1) 2021
14-106. eQEP Position-compare Unit 2022
14-107. eQEP Position-compare Event Generation Points 2023
14-108. eQEP Position-compare Sync Output Pulse Stretcher 2023
14-109. eQEP Edge Capture Unit 2025
14-110. Unit Position Event for Low Speed Measurement (QCAPCTL[UPPS] = 0010) 2025
14-111. eQEP Edge Capture Unit - Timing Details 2026
14-112. eQEP Watchdog Timer 2027
14-113. eQEP Unit Time Base 2028
14-114. EQEP Interrupt Generation 2028
14-115. eQEP Position Counter Register (QPOSCNT) 2030
14-116. eQEP Position Counter Initialization Register (QPOSINIT) 2030
14-117. eQEP Maximum Position Count Register (QPOSMAX) 2030
14-118. eQEP Position-Compare Register (QPOSCMP) 2031
14-119. eQEP Index Position Latch Register (QPOSILAT) 2031
14-120. eQEP Strobe Position Latch Register (QPOSSLAT) 2031
14-121. eQEP Position Counter Latch Register (QPOSLAT) 2032
14-122. eQEP Unit Timer Register (QUTMR) 2032
14-123. eQEP Unit Period Register (QUPRD) 2032
14-124. eQEP Watchdog Timer Register (QWDTMR) 2033
14-125. eQEP Watchdog Period Register (QWDPRD) 2033
14-126. QEP Decoder Control Register (QDECCTL) 2034
14-127. eQEP Control Register (QEPCTL) 2035
14-128. eQEP Capture Control Register (QCAPCTL) 2037
14-129. eQEP Position-Compare Control Register (QPOSCTL) 2038
14-130. eQEP Interrupt Enable Register (QEINT) 2039
14-131. eQEP Interrupt Flag Register (QFLG) 2040
14-132. eQEP Interrupt Clear Register (QCLR) 2041
14-133. eQEP Interrupt Force Register (QFRC) 2043
14-134. eQEP Status Register (QEPSTS) 2044
14-135. eQEP Capture Timer Register (QCTMR) 2045

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14-136. eQEP Capture Period Register (QCPRD) 2045
14-137. eQEP Capture Timer Latch Register (QCTMRLAT) 2045
14-138. eQEP Capture Period Latch Register (QCPRDLAT) 2046
14-139. eQEP Revision ID Register (REVID) 2046
15-1. USB Integration 2051
15-2. USB GPIO Integration 2052
15-3. CPU Actions at Transfer Phases 2060
15-4. Sequence of Transfer 2061
15-5. Flow Chart of Setup Stage of a Control Transfer in Peripheral Mode 2063
15-6. Flow Chart of Transmit Data Stage of a Control Transfer in Peripheral Mode 2064
15-7. Flow Chart of Receive Data Stage of a Control Transfer in Peripheral Mode 2065
15-8. Flow Chart of Setup Stage of a Control Transfer in Host Mode 2076
15-9. Flow Chart of Data Stage (IN Data Phase) of a Control Transfer in Host Mode 2077
15-10. Flow Chart of Data Stage (OUT Data Phase) of a Control Transfer in Host Mode 2079
15-11. Flow Chart of Status Stage of Zero Data Request or Write Request of a Control Transfer in Host Mode 2080
15-12. Chart of Status Stage of a Read Request of a Control Transfer in Host Mode 2082
15-13. Packet Descriptor Layout 2092
15-14. Buffer Descriptor (BD) Layout 2095
15-15. Teardown Descriptor Layout 2097
15-16. Relationship Between Memory Regions and Linking RAM 2102
15-17. High-level Transmit and Receive Data Transfer Example 2107
15-18. Transmit Descriptors and Queue Status Configuration 2109
15-19. Transmit USB Data Flow Example (Initialization) 2110
15-20. Receive Buffer Descriptors and Queue Status Configuration 2112
15-21. Receive USB Data Flow Example (Initialization) 2113
15-22. USBSS Revision Register (REVREG) 2117
15-23. USBSS SYSCONFIG Register (SYSCONFIG) 2118
15-24. USBSS End of Interrupt Register (EOI) 2119
15-25. USBSS IRQ_STATUS_RAW (IRQSTATRAW) 2120
15-26. USBSS IRQ_STATUS (IRQSTAT) 2121
15-27. USBSS IRQ_ENABLE_SET Register (IRQENABLER) 2122
15-28. USBSS IRQ_ENABLE_CLR Register (IRQCLEARR) 2123
15-29. USBSS IRQ_DMA_THRESHOLD_TX0_0 Register (IRQDMATHOLDTX00) 2124
15-30. USBSS IRQ_DMA_THRESHOLD_TX0_1 Register (IRQDMATHOLDTX01) 2124
15-31. USBSS IRQ_DMA_THRESHOLD_TX0_2 Register (IRQDMATHOLDTX02) 2125
15-32. USBSS IRQ_DMA_THRESHOLD_TX0_3 Register (IRQDMATHOLDTX03) 2125
15-33. USBSS IRQ_DMA_THRESHOLD_RX0_0 Register (IRQDMATHOLDRX00) 2126
15-34. USBSS IRQ_DMA_THRESHOLD_RX0_1 Register (IRQDMATHOLDRX01) 2126
15-35. USBSS IRQ_DMA_THRESHOLD_RX0_2 Register (IRQDMATHOLDRX02) 2127
15-36. USBSS IRQ_DMA_THRESHOLD_RX0_3 Register (IRQDMATHOLDRX03) 2127
15-37. USBSS IRQ_DMA_THRESHOLD_TX1_0 Register (IRQDMATHOLDTX10) 2128
15-38. USBSS IRQ_DMA_THRESHOLD_TX1_1 Register (IRQDMATHOLDTX11) 2128
15-39. USBSS IRQ_DMA_THRESHOLD_TX1_2 Register (IRQDMATHOLDTX12) 2129
15-40. USBSS IRQ_DMA_THRESHOLD_TX1_3 Register (IRQDMATHOLDTX13) 2129
15-41. USBSS IRQ_DMA_THRESHOLD_RX1_0 Register (IRQDMATHOLDRX10) 2130
15-42. USBSS IRQ_DMA_THRESHOLD_RX1_1 Register (IRQDMATHOLDRX11) 2130
15-43. USBSS IRQ_DMA_THRESHOLD_RX1_2 Register (IRQDMATHOLDRX12) 2131
15-44. USBSS IRQ_DMA_THRESHOLD_RX1_3 Register (IRQDMATHOLDRX13) 2131
15-45. USBSS IRQ_DMA_ENABLE_0 Register (IRQDMAENABLE0) 2132

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15-46. USBSS IRQ_DMA_ENABLE_1 Register (IRQDMAENABLE1) 2134
15-47. USBSS IRQ_FRAME_THRESHOLD_TX0_0 Register (IRQFRAMETHOLD00) 2136
15-48. USBSS IRQ_FRAME_THRESHOLD_TX0_1 Register (IRQFRAMETHOLD01) 2136
15-49. USBSS IRQ_FRAME_THRESHOLD_TX0_2 Register (IRQFRAMETHOLD02) 2137
15-50. USBSS IRQ_FRAME_THRESHOLD_TX0_3 Register (IRQFRAMETHOLD03) 2137
15-51. USBSS IRQ_FRAME_THRESHOLD_RX0_0 Register (IRQFRAMETHOLDRX00) 2138
15-52. USBSS IRQ_FRAME_THRESHOLD_RX0_1 Register (IRQFRAMETHOLDRX01) 2138
15-53. USBSS IRQ_FRAME_THRESHOLD_RX0_2 Register (IRQFRAMETHOLDRX02) 2139
15-54. USBSS IRQ_FRAME_THRESHOLD_RX0_3 Register (IRQFRAMETHOLDRX03) 2139
15-55. USBSS IRQ_FRAME_THRESHOLD_TX1_0 Register (IRQFRAMETHOLD10) 2140
15-56. USBSS IRQ_FRAME_THRESHOLD_TX1_1 Register (IRQFRAMETHOLD11) 2140
15-57. USBSS IRQ_FRAME_THRESHOLD_TX1_2 Register (IRQFRAMETHOLD12) 2141
15-58. USBSS IRQ_FRAME_THRESHOLD_TX1_3 Register (IRQFRAMETHOLD13) 2141
15-59. USBSS IRQ_FRAME_THRESHOLD_RX1_0 Register (IRQFRAMETHOLDRX10) 2142
15-60. USBSS IRQ_FRAME_THRESHOLD_RX1_1 Register (IRQFRAMETHOLDRX11) 2142
15-61. USBSS IRQ_FRAME_THRESHOLD_RX1_2 Register (IRQFRAMETHOLDRX12) 2143
15-62. USBSS IRQ_FRAME_THRESHOLD_RX1_3 Register (IRQFRAMETHOLDRX13) 2143
15-63. USBSS IRQ_FRAME_ENABLE_0 Register (IRQFRAMEENABLE0) 2144
15-64. USBSS IRQ_FRAME_ENABLE_1 Register (IRQFRAMEENABLE1) 2146
15-65. USB0 Revision Register (USB0REV) 2149
15-66. USB0 Control Register (USB0CTRL) 2150
15-67. USB0 Status Register (USB0STAT) 2151
15-68. USB0 IRQ_MERGED_STATUS Register (USB0IRQMSTAT) 2151
15-69. USB0 IRQ_EOI Register (USB0IRQEOI) 2152
15-70. USB0 IRQ_STATUS_RAW_0 Register (USB0IRQSTATRAW0) 2153
15-71. USB0 IRQ_STATUS_RAW_1 Register (USB0IRQSTATRAW1) 2155
15-72. USB0 IRQ_STATUS_0 Register (USB0IRQSTAT0) 2157
15-73. USB0 IRQ_STATUS_1 Register (USB0IRQSTAT1) 2159
15-74. USB0 IRQ_ENABLE_SET_0 Register (USB0IRQENABLESET0) 2161
15-75. USB0 IRQ_ENABLE_SET_1 Register (USB0IRQENABLESET1) 2163
15-76. USB0 IRQ_ENABLE_CLR_0 Register (USB0IRQENABLECLR0) 2165
15-77. USB0 IRQ_ENABLE_CLR_1 Register (USB0IRQENABLECLR1) 2167
15-78. USB0 Tx Mode Register (USB0TXMODE) 2169
15-79. USB0 Rx Mode Register (USB0RXMODE) 2171
15-80. USB0 Generic RNDIS EPn Size Register (USB0GENRNDISEPn) 2173
15-81. USB0 Auto Req Register (USB0AUTOREQ) 2174
15-82. USB0 SRP Fix Time Register (USB0SRPFIXTIME) 2177
15-83. USB0 Teardown Register (USB0TDOWN) 2177
15-84. USB0 PHY UTMI Register (USB0UTMI) 2178
15-85. USB0 MGC UTMI Loopback Register (USB0UTMILB) 2179
15-86. USB0 Mode Register (USB0MODE) 2180
15-87. USB1 Revision Register (USB1REV) 2182
15-88. USB1 Control Register (USB1CTRL) 2183
15-89. USB1 Status Register (USB1STAT) 2184
15-90. USB1 IRQ_MERGED_STATUS Register (USB1IRQMSTAT) 2184
15-91. USB1 IRQ_EOI Register (USB1IRQEOI) 2185
15-92. USB1 IRQ_STATUS_RAW_0 Register (USB1IRQSTATRAW0) 2186
15-93. USB1 IRQ_STATUS_0 Register (USB1IRQSTAT0) 2188
15-94. USB1 IRQ_STATUS_1 Register (USB1IRQSTAT1) 2190

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15-95. USB1 IRQ_ENABLE_SET_0 Register (USB1IRQENABLESET0) 2192
15-96. USB1 IRQ_ENABLE_SET_1 Register (USB1IRQENABLESET1) 2194
15-97. USB1 IRQ_ENABLE_CLR_0 Register (USB1IRQENABLECLR0) 2196
15-98. USB1 IRQ_ENABLE_CLR_1 Register (USB1IREENABLECLR1) 2198
15-99. USB1 Tx Mode Register (USB1TXMODE) 2200
15-100. USB1 Rx Mode Register (USB1RXMODE) 2202
15-101. USB1 Generic RNDIS EP N Size Register (USB1GENRNDISEPn) 2204
15-102. USB1 Auto Req Register (USB1AUTOREQ) 2205
15-103. USB1 SRP Fix Time Register (USB1SRPFIXTIME) 2208
15-104. USB1 Teardown Register (USB1TDOWN) 2208
15-105. USB1 PHY UTMI Register (USB1UTMI) 2209
15-106. USB1 MGC UTMI Loopback Register (USB1UTMILB) 2210
15-107. USB1 Mode Register (USB1MODE) 2211
15-108. CPPI DMA Revision Register (DMAREVID) 2213
15-109. CPPI DMA Teardown Free Descriptor Queue Control Register (TDFDQ) 2213
15-110. CPPI DMA Emulation Control Register (DMAEMU) 2214
15-111. CPPI Mem1 Base Address Register (DMAMEM1BA) 2214
15-112. CPPI Mem1 Mask Address Register (DMAMEM1MASK) 2215
15-113. Tx Channel N Global Configuration Register (TXGCRn) 2215
15-114. Rx Channel N Global Configuration Register (RXGCRn) 2216
15-115. Rx Channel N Host Packet Configuration Register A (RXHPCRAn) 2217
15-116. Rx Channel N Host Packet Configuration Register B (RXHPCRBn) 2218
15-117. CPPI DMA Scheduler Control Register (DMA_SCHED_CTRL) 2219
15-118. CPPI DMA Scheduler Table Word N (WORDn) 2220
15-119. Queue Manager Revision Register (QMGRREVID) 2223
15-120. Queue Manager Reset Register (QMGRRST) 2223
15-121. Queue Manager Queue Diversion Register (DIVERSION) 2224
15-122. Queue Manager Free Descriptor/Buffer Starvation Count Register 0 (FDBSC0) 2225
15-123. Queue Manager Free Descriptor/Buffer Starvation Count Register 1 (FDBSC1) 2225
15-124. Queue Manager Free Descriptor/Buffer Starvation Count Register 2 (FDBSC2) 2227
15-125. Queue Manager Free Descriptor/Buffer Starvation Count Register 3 (FDBSC3) 2227
15-126. Queue Manager Free Descriptor/Buffer Starvation Count Register 4 (FDBSC4) 2229
15-127. Queue Manager Free Descriptor/Buffer Starvation Count Register 5 (FDBSC5) 2229
15-128. Queue Manager Free Descriptor/Buffer Starvation Count Register 6 (FDBSC6) 2231
15-129. Queue Manager Free Descriptor/Buffer Starvation Count Register 7 (FDBSC7) 2231
15-130. Queue Manager Linking RAM Region 0 Base Address Register (LRAM0BASE) 2233
15-131. Queue Manager Linking RAM Region 0 Size Register (LRAM0SIZE) 2233
15-132. Queue Manager Linking RAM Region 1 Base Address Register (LRAM1BASE) 2234
15-133. Queue Manager Queue Pending Register 0 (PEND0) 2234
15-134. Queue Manager Queue Pending Register 1 (PEND1) 2235
15-135. Queue Manager Queue Pending Register 2 (PEND2) 2235
15-136. Queue Manager Queue Pending Register 3 (PEND3) 2235
15-137. Queue Manager Queue Pending Register 4 (PEND4) 2236
15-138. Queue Manager Memory Region R Base Address Register (QMEMRBASEr) 2236
15-139. Queue Manager Memory Region R Control Register (QMEMRCTRLr) 2237
15-140. Queue Manager Queue N Register A (CTRLAn) 2238
15-141. Queue Manager Queue N Register B (CTRLBn) 2238
15-142. Queue Manager Queue N Register C (CTRLCn) 2239
15-143. Queue Manager Queue N Register D (CTRLDn) 2239

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15-144. Queue Manager Queue N Register A (QSTATAn) 2240
15-145. Queue Manager Queue N Status Register B (QSTATBn) 2240
15-146. Queue Manager Queue N Status Register C (QSTATCn) 2241
15-147. Function Address Register (USBn_FADDR) 2243
15-148. Power Management Register (USBn_POWER) 2243
15-149. Interrupt Register for Endpoint 0 Plus Transmit Endpoints 1 to 15 (USBn_INTRTX) 2244
15-150. Interrupt Register for Receive Endpoints 1 to 15 (INTRRX) 2245
15-151. Interrupt Enable Register for INTRTX (INTRTXE) 2246
15-152. Interrupt Enable Register for INTRRX (INTRRXE) 2247
15-153. Interrupt Register for Common USB Interrupts (USBn_INTRUSB) 2248
15-154. Interrupt Enable Register for INTRUSB (USBn_INTRUSBE) 2248
15-155. Frame Number Register (USBn_FRAME) 2249
15-156. Index Register for Selecting the Endpoint Status and Control Registers (USBn_INDEX) 2249
15-157. Register to Enable the USB 2.0 Test Modes (USBn_TESTMODE) 2250
15-158. Maximum Packet Size for Peripheral/Host Transmit Endpoint Register (USBn_TXMAXP) 2251
15-159. Control Status Register for Endpoint 0 in Peripheral Mode (USBn_PERI_CSR0) 2252
15-160. Control Status Register for Endpoint 0 in Host Mode (USBn_HOST_CSR0) 2253
15-161. Control Status Register for Peripheral Transmit Endpoint (USBn_PERI_TXCSR) 2254
15-162. Control Status Register for Host Transmit Endpoint (USBn_HOST_TXCSR) 2255
15-163. Maximum Packet Size for Peripheral/Host Receive Endpoint Register (USBn_RXMAXP) 2256
15-164. Control Status Register for Peripheral Receive Endpoint (USBn_PERI_RXCSR) 2257
15-165. Control Status Register for Host Receive Endpoint (USBn_HOST_RXCSR) 2259
15-166. Count 0 Register (USBn_COUNT0) 2260
15-167. Receive Count Register (USBn_RXCOUNT) 2261
15-168. Type Register (Host Mode Only) (USBn_HOST_TYPE0) 2261
15-169. Transmit Type Register (Host Mode Only) (USBn_HOST_TXTYPE) 2262
15-170. NAKLimit0 Register (Host Mode Only) (USBn_HOST_NAKLIMIT0) 2263
15-171. Transmit Interval Register (Host Mode Only) (USBn_HOST_TXINTERVAL) 2264
15-172. Receive Type Register (Host Mode Only) (USBn_HOST_RXTYPE) 2265
15-173. Receive Interval Register (Host Mode Only) (USBn_HOST_RXINTERVAL) 2266
15-174. Configuration Data Register (USBn_CONFIGDATA) 2267
15-175. Transmit and Receive FIFO Register for Endpoint 0 - 15 (USBn_FIFO0 – USBn_FIFO15) 2268
15-176. Device Control Register (USBn_DEVCTL) 2269
15-177. Transmit Endpoint FIFO Size Register (USBn_TXFIFOSZ) 2270
15-178. Receive Endpoint FIFO Size Register (USBn_RXFIFOSZ) 2271
15-179. Transmit Endpoint FIFO Address Register (USBn_TXFIFOADDR) 2271
15-180. Receive Endpoint FIFO Address Register (USBn_RXFIFOADDR) 2272
15-181. Hardware Version Register (USBn_HWVERS) 2272
15-182. Transmit Function Address Register (USBn_TXFUNCADDRm) 2273
15-183. Transmit Hub Address Register (USBn_TXHUBADDRm) 2274
15-184. Transmit Hub Port Register (USBn_TXHUBPORTm) 2274
15-185. Receive Function Address Register (USBn_RXFUNCADDRm) 2275
15-186. Receive Hub Address Register (USBn_RXHUBADDRm) 2275
15-187. Receive Hub Port Register (USBn_RXHUBPORTm) 2277
16-1. REV Register 2286
16-2. SYSCONFIG Register 2287
16-3. SYSTATUS Register 2288
16-4. LOCK_REG_0 Register 2289
16-5. LOCK_REG_1 Register 2290

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16-6. LOCK_REG_2 Register 2291
16-7. LOCK_REG_3 Register 2292
16-8. LOCK_REG_4 Register 2293
16-9. LOCK_REG_5 Register 2294
16-10. LOCK_REG_6 Register 2295
16-11. LOCK_REG_7 Register 2296
16-12. LOCK_REG_8 Register 2297
16-13. LOCK_REG_9 Register 2298
16-14. LOCK_REG_10 Register 2299
16-15. LOCK_REG_11 Register 2300
16-16. LOCK_REG_12 Register 2301
16-17. LOCK_REG_13 Register 2302
16-18. LOCK_REG_14 Register 2303
16-19. LOCK_REG_15 Register 2304
16-20. LOCK_REG_16 Register 2305
16-21. LOCK_REG_17 Register 2306
16-22. LOCK_REG_18 Register 2307
16-23. LOCK_REG_19 Register 2308
16-24. LOCK_REG_20 Register 2309
16-25. LOCK_REG_21 Register 2310
16-26. LOCK_REG_22 Register 2311
16-27. LOCK_REG_23 Register 2312
16-28. LOCK_REG_24 Register 2313
16-29. LOCK_REG_25 Register 2314
16-30. LOCK_REG_26 Register 2315
16-31. LOCK_REG_27 Register 2316
16-32. LOCK_REG_28 Register 2317
16-33. LOCK_REG_29 Register 2318
16-34. LOCK_REG_30 Register 2319
16-35. LOCK_REG_31 Register 2320
17-1. MMCHS Module SDIO Application 2323
17-2. MMCHS SD (4-bit) Card Application 2323
17-3. MMCHS Module MMC Application 2323
17-4. MMC/SD 0/2 Connectivity to an MMC/SD Card 2326
17-5. MMC/SD 1 Connectivity to an MMC/SD Card 2327
17-6. Sequential Read Operation (MMC Cards Only) 2329
17-7. Sequential Write Operation (MMC Cards Only) 2329
17-8. Multiple Block Read Operation (MMC Cards Only) 2330
17-9. Multiple Block Write Operation (MMC Cards Only) 2330
17-10. Command Token Format 2331
17-11. 48-Bit Response Packet (R1, R3, R4, R5, R6) 2331
17-12. 136-Bit Response Packet (R2) 2331
17-13. Data Packet for Sequential Transfer (1-Bit) 2332
17-14. Data Packet for Block Transfer (1-Bit) 2332
17-15. Data Packet for Block Transfer (4-Bit) 2332
17-16. Data Packet for Block Transfer (8-Bit) 2333
17-17. DMA Receive Mode 2340
17-18. DMA Transmit Mode 2341
17-19. Buffer Management for a Write 2343

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17-20. Buffer Management for a Read 2344
17-21. Busy Timeout for R1b, R5b Responses 2347
17-22. Busy Timeout After Write CRC Status 2347
17-23. Write CRC Status Timeout 2348
17-24. Read Data Timeout 2348
17-25. Boot Acknowledge Timeout When Using CMD0 2349
17-26. Boot Acknowledge Timeout When CMD Held Low 2349
17-27. Auto CMD12 Timing During Write Transfer 2351
17-28. Auto Command 12 Timings During Read Transfer 2352
17-29. Output Driven on Falling Edge 2354
17-30. Output Driven on Rising Edge 2355
17-31. Boot Mode With CMD0 2356
17-32. Boot Mode With CMD Line Tied to 0 2356
17-33. MMC/SD/SDIO Controller Software Reset Flow 2360
17-34. MMC/SD/SDIO Controller Bus Configuration Flow 2361
17-35. MMC/SD/SDIO Controller Card Identification and Selection - Part 1 2362
17-36. MMC/SD/SDIO Controller Card Identification and Selection - Part 2 2363
17-37. SD_HL_REV Register 2365
17-38. SD_HL_HWINFO Register 2365
17-39. SD_HL_SYSCONFIG Register 2365
17-40. System Configuration Register (SD_SYSCONFIG) 2366
17-41. System Status Register (SD_SYSSTATUS) 2368
17-42. Card Status Response Error (SD_CSRE) 2368
17-43. System Test Register (SD_SYSTEST) 2369
17-44. Configuration Register (SD_CON) 2372
17-45. Power Counter Register (SD_PWCNT) 2375
17-46. Card Status Response Error (SD_SDMASA) 2375
17-47. Transfer Length Configuration Register (SD_BLK) 2376
17-48. Command Argument Register (SD_ARG) 2377
17-49. Command and Transfer Mode Register (SD_CMD) 2377
17-50. Command Response[31:0] Register (SD_RSP10) 2381
17-51. Command Response[63:32] Register (SD_RSP32) 2381
17-52. Command Response[95:64] Register (SD_RSP54) 2382
17-53. Command Response[127:96] Register (SD_RSP76) 2382
17-54. Data Register (SD_DATA) 2383
17-55. Present State Register (SD_PSTATE) 2384
17-56. Control Register (SD_HCTL) 2387
17-57. SD System Control Register (SD_SYSCTL) 2390
17-58. Interrupt Status Register (SD_STAT) 2392
17-59. Interrupt SD Enable Register (SD_IE) 2397
17-60. Interrupt Signal Enable Register (SD_ISE) 2400
17-61. Auto CMD12 Error Status Register (SD_AC12) 2403
17-62. Capabilities Register (SD_CAPA) 2404
17-63. Maximum Current Capabilities Register (SD_CUR_CAPA) 2406
17-64. Interrupt Signal Enable Register (SD_ISE) 2407
17-65. ADMA Error Status Register (SD_ADMAES) 2409
17-66. ADMA System Address Low Bits (SD_ADMASAL) 2410
17-67. ADMA System Address High Bits Register (SD_ADMASAH) 2410
17-68. Versions Register (SD_REV) 2411

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18-1. UART/IrDA Module — UART Application 2416
18-2. UART/IrDA Module — IrDA/CIR Application 2416
18-3. UART/IrDA/CIR Functional Specification Block Diagram 2420
18-4. FIFO Management Registers 2426
18-5. RX FIFO Interrupt Request Generation 2428
18-6. TX FIFO Interrupt Request Generation 2428
18-7. Receive FIFO DMA Request Generation (32 Characters) 2430
18-8. Transmit FIFO DMA Request Generation (56 Spaces) 2430
18-9. Transmit FIFO DMA Request Generation (8 Spaces) 2431
18-10. Transmit FIFO DMA Request Generation (1 Space) 2432
18-11. Transmit FIFO DMA Request Generation Using Direct TX DMA Threshold Programming. (Threshold = 3; Spaces = 8) 2433
18-12. DMA Transmission 2433
18-13. DMA Reception 2434
18-14. Baud Rate Generation 2440
18-15. Baud Rate Generator 2446
18-16. CIR Mode Block Components 2450
18-17. Receiver Holding Register (RHR) 2463
18-18. Transmit Holding Register (THR) 2463
18-19. UART Interrupt Enable Register (IER) 2464
18-20. IrDA Interrupt Enable Register (IER) 2465
18-21. CIR Interrupt Enable Register (IER) 2466
18-22. UART Interrupt Identification Register (IIR) 2467
18-23. IrDA Interrupt Identification Register (IIR) 2468
18-24. CIR Interrupt Identification Register (IIR) 2469
18-25. FIFO Control Register (FCR) 2470
18-26. Line Control Register (LCR) 2471
18-27. Modem Control Register (MCR) 2472
18-28. UART Line Status Register (LSR) 2473
18-29. IrDA Line Status Register (LSR) 2474
18-30. CIR Line Status Register (LSR) 2475
18-31. Modem Status Register (MSR) 2476
18-32. Transmission Control Register (TCR) 2477
18-33. Scratchpad Register (SPR) 2477
18-34. Trigger Level Register (TLR) 2478
18-35. Mode Definition Register 1 (MDR1) 2479
18-36. Mode Definition Register 2 (MDR2) 2480
18-37. Status FIFO Line Status Register (SFLSR) 2481
18-38. RESUME Register 2481
18-39. Status FIFO Register Low (SFREGL) 2482
18-40. Status FIFO Register High (SFREGH) 2482
18-41. BOF Control Register (BLR) 2483
18-42. Auxiliary Control Register (ACREG) 2484
18-43. Supplementary Control Register (SCR) 2485
18-44. Supplementary Status Register (SSR) 2486
18-45. BOF Length Register (EBLR) 2487
18-46. Module Version Register (MVR) 2488
18-47. System Configuration Register (SYSC) 2489
18-48. System Status Register (SYSS) 2489

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18-49. Wake-Up Enable Register (WER) 2490
18-50. Carrier Frequency Prescaler Register (CFPS) 2491
18-51. Divisor Latches Low Register (DLL) 2492
18-52. Divisor Latches High Register (DLH) 2492
18-53. Enhanced Feature Register (EFR) 2493
18-54. XON1/ADDR1 Register 2494
18-55. XON2/ADDR2 Register 2494
18-56. XOFF1 Register 2495
18-57. XOFF2 Register 2495
18-58. Transmit Frame Length Low Register (TXFLL) 2496
18-59. Transmit Frame Length High Register (TXFLH) 2496
18-60. Received Frame Length Low Register (RXFLL) 2497
18-61. Received Frame Length High Register (RXFLH) 2497
18-62. UART Autobauding Status Register (UASR) 2498
19-1. Timer Block Diagram 2501
19-2. TCRR Timing Value 2504
19-3. Capture Wave Example for CAPT_MODE = 0 2505
19-4. Capture Wave Example for CAPT_MODE = 1 2506
19-5. Timing Diagram of Pulse-Width Modulation with SCPWM = 0 2507
19-6. Timing Diagram of Pulse-Width Modulation with SCPWM = 1 2508
19-7. Identification Register (TIDR) Register 2514
19-8. Timer OCP Configuration Register (TIOCP_CFG) 2515
19-9. Timer IRQ EOI Register (IRQ_EOI) 2516
19-10. Timer IRQSTATUS Raw Register (IRQSTATUS_RAW) 2517
19-11. Timer IRQSTATUS Register (IRQSTATUS) 2518
19-12. 3Timer IRQENABLE Set Register (IRQENABLE_SET) 2519
19-13. Timer IRQENABLE Clear Register (IRQENABLE_CLR) 2520
19-14. Timer IRQ Wakeup Enable Register (IRQWAKEEN) 2521
19-15. Timer Control Register (TCLR) 2521
19-16. Timer Counter Register (TCRR) 2523
19-17. Timer Load Register (TLDR) 2523
19-18. Timer Trigger Register (TTGR) 2524
19-19. Timer Write Posted Status Register (TWPS) 2524
19-20. Timer Match Register (TMAR) 2525
19-21. Timer Capture Register (TCAR1) 2525
19-22. Timer Synchronous Interface Control Register (TSICR) 2526
19-23. Timer Capture Register (TCAR2) 2526
19-24. Block Diagram 2527
19-25. TCRR Timing Value 2529
19-26. 1 ms Module Block Diagram 2529
19-27. Capture Wave Example for CAPT_MODE 0 2531
19-28. Capture Wave Example for CAPT_MODE 1 2531
19-29. Timing Diagram of Pulse-Width Modulation, SCPWM Bit = 0 2532
19-30. Timing Diagram of Pulse-Width Modulation, SCPWM Bit = 1 2532
19-31. Wake-up Request Generation 2533
19-32. TIDR Register 2536
19-33. TIOCP_CFG Register 2537
19-34. TISTAT Register 2538
19-35. TISR Register 2539

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19-36. TIER Register 2540
19-37. TWER Register 2541
19-38. TCLR Register 2542
19-39. TCRR Register 2543
19-40. TLDR Register 2544
19-41. TTGR Register 2545
19-42. TWPS Register 2546
19-43. TMAR Register 2547
19-44. TCAR1 Register 2548
19-45. TSICR Register 2549
19-46. TCAR2 Register 2550
19-47. TPIR Register 2551
19-48. TNIR Register 2552
19-49. TCVR Register 2553
19-50. TOCR Register 2554
19-51. TOWR Register 2555
19-52. RTC Block Diagram 2559
19-53. RTC Functional Block Diagram 2559
19-54. Kick Register State Machine Diagram 2562
19-55. Flow Control for Updating RTC Registers 2564
19-56. Compensation Illustration 2565
19-57. Seconds Register (SECONDS_REG) 2568
19-58. Minutes Register (MINUTES_REG) 2568
19-59. Hours Register (HOURS_REG) 2569
19-60. Days of the Month Register (DAYS_REG) 2569
19-61. Month Register (MONTHS_REG) 2570
19-62. Year Register (YEARS_REG) 2570
19-63. Day of the Week Register (WEEKS_REG) 2571
19-64. Alarm Second Register (ALARM_SECONDS_REG) 2571
19-65. Alarm Minute Register (ALARM_MINUTES_REG) 2572
19-66. Alarm Hour Register (ALARM_HOURS_REG) 2572
19-67. Alarm Day of the Month (ALARM_DAYS_REG) 2573
19-68. Alarm Month Register (ALARM_MONTHS_REG) 2573
19-69. Alarm Year Register (ALARM_YEARS_REG) 2574
19-70. Control Register (CTRL_REG) 2575
19-71. Status Register (STATUS_REG) 2577
19-72. Interrupt Register (INTERRUPTS_REG) 2578
19-73. Compensation (LSB) Register (COMP_LSB_REG) 2579
19-74. Compensation (MSB) Register (COMP_MSB_REG) 2580
19-75. Oscillator Register (OSC_REG) 2581
19-76. Scratch Registers (SCRATCHx_REG) 2581
19-77. Kick0 Register (KICK0R) 2582
19-78. Kick1 Register (KICK1R) 2582
19-79. RTC Revision Register (RTC_REVISION) 2583
19-80. System Configuration Register (RTC_SYSCONFIG) 2583
19-81. Wakeup Enable Register (RTC_IRQWAKEEN) 2584
19-82. 32-Bit Watchdog Timer Functional Block Diagram 2587
19-83. Watchdog Timers General Functional View 2588
19-84. WDT_WIDR Register 2595

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19-85. WDT_WDSC Register 2596
19-86. WDT_WDST Register 2596
19-87. WDT_WISR Register 2597
19-88. WDT_WIER Register 2598
19-89. WDT_WWER Register 2598
19-90. WDT_WCLR Register 2599
19-91. WDT_WCRR Register 2600
19-92. WDT_WLDR Register 2600
19-93. WDT_WTGR Register 2600
19-94. WDT_WWPS Register 2601
19-95. WDT_WDLY 2602
19-96. WDT_WSPR Register 2602
19-97. WDT_WIRQSTATRAW Register 2603
19-98. WIRQSTAT Register 2604
19-99. WDT_WIRQENSET Register 2605
19-100. WDT_WIRQENCLR Register 2606
19-101. WIRQWAKEEN Register 2607
20-1. I2C0 Integration and Bus Application 2611
20-2. I2C(1–2) Integration and Bus Application 2611
20-3. I2C Functional Block Diagram 2613
20-4. Multiple I2C Modules Connected 2614
20-5. Bit Transfer on the I2C Bus 2615
20-6. Start and Stop Condition Events 2616
20-7. I2C Data Transfer 2616
20-8. I2C Data Transfer Formats 2617
20-9. Arbitration Procedure Between Two Master Transmitters 2618
20-10. Synchronization of Two I2C Clock Generators 2618
20-11. Receive FIFO Interrupt Request Generation 2620
20-12. Transmit FIFO Interrupt Request Generation 2621
20-13. Receive FIFO DMA Request Generation 2622
20-14. Transmit FIFO DMA Request Generation (High Threshold) 2622
20-15. Transmit FIFO DMA Request Generation (Low Threshold) 2623
20-16. Module Revision Register (LOW BYTES) (I2C_REVNB_LO) 2627
20-17. Module Revision Register (HIGH BYTES) (I2C_REVNB_HI) 2628
20-18. System Configuration Register (I2C_SYSC) 2629
20-19. I2C End of Interrupt Register (I2C_EOI) 2630
20-20. I2C Status Raw Register (I2C_IRQSTATUS_RAW) 2630
20-21. I2C Status Register (I2C_IRQSTATUS) 2635
20-22. I2C Interrupt Enable Set Register (I2C_IRQENABLE_SET) 2637
20-23. I2C Interrupt Enable Clear Register (I2C_IRQENABLE_CLR) 2639
20-24. I2C Wakeup Enable Register (I2C_WE) 2641
20-25. Receive DMA Enable Set Register (I2C_DMARXENABLE_SET) 2644
20-26. Receive DMA Enable Set Register (I2C_DMATXENABLE_SET) 2644
20-27. Receive DMA Enable Set Register (I2C_DMARXENABLE_CLR) 2645
20-28. Receive DMA Enable Set Register (I2C_DMATXENABLE_CLR) 2645
20-29. Receive DMA Wakeup Register (I2C_DMARXWAKE_EN) 2646
20-30. Receive DMA Wakeup Register (I2C_DMATXWAKE_EN) 2648
20-31. System Status Register (I2C_SYSS) 2650
20-32. Buffer Configuration Register (I2C_BUF) 2650

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20-33. Data Counter Register (I2C_CNT) 2652
20-34. Data Access Register (I2C_DATA) 2653
20-35. I2C Configuration Register (I2C_CON) 2654
20-36. I2C Own Address Register (I2C_OA) 2656
20-37. I2C Own Address Register (I2C_SA) 2657
20-38. I2C Own Address Register (I2C_PSC) 2658
20-39. Clock Divider 2658
20-40. I2C SCL Low Time Register (I2C_SCLL) 2659
20-41. I2C SCL High Time Register (I2C_SCLH) 2659
20-42. System Test Register (I2C_SYSTEST) 2660
20-43. I2C Buffer Status Register (I2C_BUFSTAT) 2663
20-44. Own Address 1 (OA1) (I2C_OA1) 2664
20-45. I2C Own Address 2 Register (I2C_OA2) 2665
20-46. I2C Own Address 3 Register (I2C_OA3) 2666
20-47. Active Own Address Register (I2C_ACTOA) 2667
20-48. I2C Clock Blocking Enable Register (I2C_SBLOCK) 2668
21-1. McASP0–1 Integration 2672
21-2. Transmit Clock Generator Block Diagram 2675
21-3. Receive Clock Generator Block Diagram 2676
21-4. Frame Sync Generator Block Diagram 2677
21-5. Burst Frame Sync Mode 2679
21-6. Transmit DMA Event (AXEVT) Generation in TDM Time Slots 2681
21-7. Individual Serializer and Connections Within McASP 2686
21-8. Receive Format Unit 2687
21-9. Transmit Format Unit 2687
21-10. McASP I/O Pin Control Block Diagram 2689
21-11. Processor Service Time Upon Transmit DMA Event (AXEVT) 2691
21-12. Processor Service Time Upon Receive DMA Event (AREVT) 2692
21-13. McASP Audio FIFO (AFIFO) Block Diagram 2694
21-14. Data Flow Through Transmit Format Unit, Illustrated 2697
21-15. Data Flow Through Receive Format Unit, Illustrated 2699
21-16. Transmit Clock Failure Detection Circuit Block Diagram 2703
21-17. Receive Clock Failure Detection Circuit Block Diagram 2705
21-18. Serializers in Loopback Mode 2706
21-19. Interrupt Multiplexing 2712
21-20. Audio Mute (AMUTE) Block Diagram 2713
21-21. DMA Events in an Audio Example–Two Events (Scenario 1) 2715
21-22. DMA Events in an Audio Example–Four Events (Scenario 2) 2715
21-23. DMA Events in an Audio Example 2716
21-24. Revision Identification Register (REV) 2720
21-25. Pin Function Register (PFUNC) 2721
21-26. Pin Direction Register (PDIR) 2723
21-27. Pin Data Output Register (PDOUT) 2725
21-28. Pin Data Input Register (PDIN) 2727
21-29. Pin Data Set Register (PDSET) 2729
21-30. Pin Data Clear Register (PDCLR) 2731
21-31. Global Control Register (GBLCTL) 2733
21-32. Audio Mute Control Register (AMUTE) 2735
21-33. Digital Loopback Control Register (DLBCTL) 2737

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21-34. Digital Mode Control Register (DITCTL) 2738
21-35. Receiver Global Control Register (RGBLCTL) 2739
21-36. Receive Format Unit Bit Mask Register (RMASK) 2740
21-37. Receive Bit Stream Format Register (RFMT) 2741
21-38. Receive Frame Sync Control Register (AFSRCTL) 2743
21-39. Receive Clock Control Register (ACLKRCTL) 2744
21-40. Receive High-Frequency Clock Control Register (AHCLKRCTL) 2745
21-41. Receive TDM Time Slot Register (RTDM) 2746
21-42. Receiver Interrupt Control Register (RINTCTL) 2747
21-43. Receiver Status Register (RSTAT) 2748
21-44. Current Receive TDM Time Slot Registers (RSLOT) 2749
21-45. Receive Clock Check Control Register (RCLKCHK) 2750
21-46. Receiver DMA Event Control Register (REVTCTL) 2751
21-47. Transmitter Global Control Register (XGBLCTL) 2752
21-48. Transmit Format Unit Bit Mask Register (XMASK) 2753
21-49. Transmit Bit Stream Format Register (XFMT) 2754
21-50. Transmit Frame Sync Control Register (AFSXCTL) 2756
21-51. Transmit Clock Control Register (ACLKXCTL) 2757
21-52. Transmit High-Frequency Clock Control Register (AHCLKXCTL) 2758
21-53. Transmit TDM Time Slot Register (XTDM) 2759
21-54. Transmitter Interrupt Control Register (XINTCTL) 2760
21-55. Transmitter Status Register (XSTAT) 2761
21-56. Current Transmit TDM Time Slot Register (XSLOT) 2762
21-57. Transmit Clock Check Control Register (XCLKCHK) 2763
21-58. Transmitter DMA Event Control Register (XEVTCTL) 2764
21-59. Serializer Control Registers (SRCTL*n*) 2765
21-60. DIT Left Channel Status Registers (DITCSRA0-DITCSRA5) 2766
21-61. DIT Right Channel Status Registers (DITCSRB0-DITCSRB5) 2766
21-62. DIT Left Channel User Data Registers (DITUDRA0-DITUDRA5) 2766
21-63. DIT Right Channel User Data Registers (DITUDRB0-DITUDRB5) 2766
21-64. Transmit Buffer Registers (XBUFn) 2767
21-65. Receive Buffer Registers (RBUF*n*) 2767
21-66. Write FIFO Control Register (WFIFOCTL) 2768
21-67. Write FIFO Status Register (WFIFOSTS) 2769
21-68. Read FIFO Control Register (RFIFOCTL) 2770
21-69. Read FIFO Status Register (RFIFOSTS) 2771
22-1. DCAN Integration 2775
22-2. DCAN Block Diagram 2776
22-3. CAN Module General Initialization Flow 2778
22-4. CAN Bit-Timing Configuration 2779
22-5. CAN Core in Silent Mode 2781
22-6. CAN Core in Loopback Mode 2782
22-7. CAN Core in External Loopback Mode 2783
22-8. CAN Core in Loop Back Combined With Silent Mode 2784
22-9. CAN Interrupt Topology 1 2786
22-10. CAN Interrupt Topology 2 2786
22-11. Local Power-Down Mode Flow Diagram 2788
22-12. CPU Handling of a FIFO Buffer (Interrupt Driven) 2797
22-13. Bit Timing 2798

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22-14. The Propagation Time Segment 2799
22-15. Synchronization on Late and Early Edges 2801
22-16. Filtering of Short Dominant Spikes 2802
22-17. Structure of the CAN Core's CAN Protocol Controller 2803
22-18. Data Transfer Between IF1/IF2 Registers and Message RAM 2807
22-19. CAN Control Register (DCAN CTL) 2815
22-20. Error and Status Register (DCAN ES) 2817
22-21. Parity Error End of Interrupt Register (PARITYERR_EOI) 2817
22-22. Error Counter Register (DCAN ERRC) 2819
22-23. Bit Timing Register (DCAN BTR) 2820
22-24. Interrupt Register (DCAN INT) 2821
22-25. Test Register (DCAN TEST) 2822
22-26. Parity Error Code Register (DCAN PERR) 2823
22-27. Auto-Bus-On Time Register (DCAN ABOTR) 2824
22-28. Transmission Request X Register (DCAN TXRQ X) 2825
22-29. Transmission Request X Register (DCAN TXRQ X) 2827
22-30. Interrupt Pending X Register (DCAN INTPND X) 2829
22-31. Message Valid X Register (DCAN MSGVAL X) 2831
22-32. IF1 Command Registers (DCAN IF1CMD) 2834
22-33. IF2 Command Registers (DCAN IF2CMD) 2834
22-34. IF1 Mask Register (DCAN IF1MSK) 2837
22-35. IF2 Mask Register (DCAN IF2MSK) 2837
22-36. IF1 Arbitration Register (DCAN IF1ARB) 2838
22-37. IF2 Arbitration Register (DCAN IF2ARB) 2838
22-38. IF1 Message Control Register (DCAN IF1MCTL) 2840
22-39. IF2 Message Control Register (DCAN IF2MCTL) 2840
22-40. IF1 Data A Register (DCAN IF1DATA) 2842
22-41. IF1 Data B Register (DCAN IF1DATA) 2842
22-42. IF2 Data A Register (DCAN IF2DATA) 2842
22-43. IF2 Data B Register (DCAN IF2DATA) 2842
22-44. IF3 Observation Register (DCAN IF3OBS) 2843
22-45. IF3 Mask Register (DCAN IF3MSK) 2845
22-46. IF3 Arbitration Register (DCAN IF3ARB) 2846
22-47. IF3 Message Control Register (DCAN IF3MCTL) 2847
22-48. IF3 Data A Register (DCAN IF3DATA) 2849
22-49. IF3 Data A Register (DCAN IF3DATB) 2849
22-50. CAN TX I/O Control Register (DCAN TIOC) 2851
22-51. CAN RX IO control register (DCAN RIOC) 2853
23-1. SPI Master Application 2857
23-2. SPI Slave Application 2857
23-3. SPI Full-Duplex Transmission 2860
23-4. SPI Half-Duplex Transmission (Receive-only Slave) 2861
23-5. SPI Half-Duplex Transmission (Transmit-Only Slave) 2861
23-6. Phase and Polarity Combinations 2863
23-7. Full Duplex Single Transfer Format with PHA = 0 2864
23-8. Full Duplex Single Transfer Format With PHA = 1 2865
23-9. Continuous Transfers With SPIEN Maintained Active (Single-Data-Pin Interface Mode) 2870
23-10. Continuous Transfers With SPIEN Maintained Active (Dual-Data-Pin Interface Mode) 2870
23-11. Extended SPI Transfer With Start Bit PHA = 1 2872

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23-12. Chip-Select SPIEN Timing Controls 2873
23-13. Transmit/Receive Mode With No FIFO Used 2877
23-14. Transmit/Receive Mode With Only Receive FIFO Enabled 2877
23-15. Transmit/Receive Mode With Only Transmit FIFO Used 2878
23-16. Transmit/Receive Mode With Both FIFO Direction Used 2878
23-17. Transmit-Only Mode With FIFO Used 2879
23-18. Receive-Only Mode With FIFO Used 2879
23-19. Buffer Almost Full Level (AFL) 2880
23-20. Buffer Almost Empty Level (AEL) 2881
23-21. Master Single Channel Initial Delay 2882
23-22. 3-Pin Mode System Overview 2883
23-23. Example of SPI Slave with One Master and Multiple Slave Devices on Channel 0 2885
23-24. SPI Half-Duplex Transmission (Receive-Only Slave) 2887
23-25. SPI Half-Duplex Transmission (Transmit-Only Slave) 2888
23-26. McSPI System Configuration Register (MCSPI_SYSCONFIG) 2896
23-27. McSPI System Status Register (MCSPI_SYSSTATUS) 2897
23-28. McSPI Interrupt Status Register (MCSPI_IRQSTATUS) 2898
23-29. McSPI Interrupt Enable Register (MCSPI_IRQENABLE) 2901
23-30. McSPI System Register (MCSPI_SYST) 2903
23-31. McSPI Module Control Register (MCSPI_MODULCTRL) 2905
23-32. McSPI Channel *(i)* Configuration Register (MCSPI_CH(i)CONF) 2907
23-33. McSPI Channel *(i)* Status Register (MCSPI_CH(i)STAT) 2911
23-34. McSPI Channel *(i)* Control Register (MCSPI_CH(I)CTRL) 2912
23-35. McSPI Channel *(i)* Transmit Register (MCSPI_TX*(i)*) 2913
23-36. McSPI Channel *(i)* Receive Register (MCSPI_RX*(i)*) 2913
23-37. McSPI Transfer Levels Register (MCSPI_XFERLEVEL) 2914
24-1. GPIO0 Module Integration 2917
24-2. GPIO[1–3] Module Integration 2917
24-3. Interrupt Request Generation 2922
24-4. Wake-Up Request Generation 2924
24-5. Write @ GPIO_CLEARDATAOUT Register Example 2927
24-6. Write @ GPIO_SETIRQENABLEx Register Example 2928
24-7. General-Purpose Interface Used as a Keyboard Interface 2929
24-8. GPIO_REVISION Register 2931
24-9. GPIO_SYSCONFIG Register 2932
24-10. GPIO_EOI Register 2933
24-11. GPIO_IRQSTATUS_RAW_n Register 2934
24-12. GPIO_IRQSTATUS_n Register 2934
24-13. GPIO_IRQSTATUS_SET_n Register 2935
24-14. GPIO_IRQSTATUS_CLR_n Register 2935
24-15. GPIO_IRQWAKEN_n Register 2936
24-16. GPIO_SYSSTATUS Register 2937
24-17. GPIO_CTRL Register 2938
24-18. GPIO_OE Register 2938
24-19. GPIO_DATAIN Register 2939
24-20. GPIO_DATAOUT Register 2939
24-21. GPIO_LEVELDETECT0 Register 2940
24-22. GPIO_LEVELDETECT1 Register 2940
24-23. GPIO_RISINGDETECT Register 2941

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

24-24. GPIO_FALLINGDETECT Register 2941
24-25. GPIO_DEBOUNCENABLE Register 2942
24-26. GPIO_DEBOUNCINGTIME Register 2942
24-27. GPIO_CLEARDATAOUT Register 2943
24-28. GPIO_SETDATAOUT Register 2943
25-1. Public ROM Code Architecture 2946
25-2. Public ROM Code Boot Procedure 2947
25-3. ROM Memory Map 2947
25-4. Public RAM Memory Map 2948
25-5. ROM Code Startup Sequence 2950
25-6. ROM Code Booting Procedure 2951
25-7. Fast External Boot 2953
25-8. Memory Booting 2953
25-9. GPMC XIP Timings 2955
25-10. Image Shadowing on GP Device 2956
25-11. GPMC NAND Timings 2957
25-12. NAND Device Detection 2960
25-13. NAND Invalid Blocks Detection 2960
25-14. NAND Read Sector Procedure 2960
25-15. ECC Data Mapping for 2 KB Page and 8b BCH Encoding 2961
25-16. ECC Data Mapping for 4 KB Page and 16b BCH Encoding 2961
25-17. MMC/SD Booting 2962
25-18. MMC/SD Detection Procedure 2962
25-19. SD/MMC Booting, Get Booting File 2964
25-20. MBR Detection Procedure 2965
25-21. MBR, Get Partition 2965
25-22. FAT Detection Procedure 2967
25-23. Peripheral Booting Procedure 2971
25-24. USB Initialization Procedure 2974
25-25. Image Transfer for USB Boot 2974
25-26. Image Formats on GP and HS Devices 2975
25-27. Image Structure 2976
25-28. TOC Structure 2976
25-29. HS Image Authentication Process 2977
25-30. Wakeup Booting by ROM 2979

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

List of Tables

2-1. ARM Cortex A8 Interrupts 132
2-2. ARM Cortex M3 Wakeup Processor Interrupts 135
2-3. PRUSS1 Interrupts 137
2-4. EDMA Events 139
2-5. Crypto DMA Events 142
2-6. Timer & eCAP Event Capture 144
2-7. INTCPS REGISTERS 145
2-8. INTCPS_REVISION Register Field Descriptions 151
2-9. INTCPS_FIQ_PRIORITY Register Field Descriptions 152
2-10. INTCPS_THRESHOLD Register Field Descriptions 153
2-11. INTCPS_ITR0 Register Field Descriptions 154
2-12. INTCPS_MIR0 Register Field Descriptions 155
2-13. INTCPS_MIR_CLEAR0 Register Field Descriptions 156
2-14. INTCPS_MIR_SET0 Register Field Descriptions 157
2-15. INTCPS_ISR_SET0 Register Field Descriptions 158
2-16. INTCPS_ISR_CLEAR0 Register Field Descriptions 159
2-17. INTCPS_PENDING_IRQ0 Register Field Descriptions 160
2-18. INTCPS_PENDING_FIQ0 Register Field Descriptions 161
2-19. INTCPS_SYSCONFIG Register Field Descriptions 162
2-20. INTCPS_ITR1 Register Field Descriptions 163
2-21. INTCPS_MIR1 Register Field Descriptions 164
2-22. INTCPS_MIR_CLEAR1 Register Field Descriptions 165
2-23. INTCPS_MIR_SET1 Register Field Descriptions 166
2-24. INTCPS_ISR_SET1 Register Field Descriptions 167
2-25. INTCPS_ISR_CLEAR1 Register Field Descriptions 168
2-26. INTCPS_PENDING_IRQ1 Register Field Descriptions 169
2-27. INTCPS_PENDING_FIQ1 Register Field Descriptions 170
2-28. INTCPS_ITR2 Register Field Descriptions 171
2-29. INTCPS_MIR2 Register Field Descriptions 172
2-30. INTCPS_SYSSTATUS Register Field Descriptions 173
2-31. INTCPS_MIR_CLEAR2 Register Field Descriptions 174
2-32. INTCPS_MIR_SET2 Register Field Descriptions 175
2-33. INTCPS_ISR_SET2 Register Field Descriptions 176
2-34. INTCPS_ISR_CLEAR2 Register Field Descriptions 177
2-35. INTCPS_PENDING_IRQ2 Register Field Descriptions 178
2-36. INTCPS_PENDING_FIQ2 Register Field Descriptions 179
2-37. INTCPS_ITR3 Register Field Descriptions 180
2-38. INTCPS_MIR3 Register Field Descriptions 181
2-39. INTCPS_MIR_CLEAR3 Register Field Descriptions 182
2-40. INTCPS_MIR_SET3 Register Field Descriptions 183
2-41. INTCPS_ISR_SET3 Register Field Descriptions 184
2-42. INTCPS_ISR_CLEAR3 Register Field Descriptions 185
2-43. INTCPS_PENDING_IRQ3 Register Field Descriptions 186
2-44. INTCPS_PENDING_FIQ3 Register Field Descriptions 187
2-45. INTCPS_ILR0 Register Field Descriptions 188
2-46. INTCPS_ILR1 Register Field Descriptions 189
2-47. INTCPS_ILR2 Register Field Descriptions 190

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2-48. INTCPS_ILR3 Register Field Descriptions 191
2-49. INTCPS_ILR4 Register Field Descriptions 192
2-50. INTCPS_ILR5 Register Field Descriptions 193
2-51. INTCPS_ILR6 Register Field Descriptions 194
2-52. INTCPS_ILR7 Register Field Descriptions 195
2-53. INTCPS_ILR8 Register Field Descriptions 196
2-54. INTCPS_ILR9 Register Field Descriptions 197
2-55. INTCPS_ILR10 Register Field Descriptions 198
2-56. INTCPS_ILR11 Register Field Descriptions 199
2-57. INTCPS_ILR12 Register Field Descriptions 200
2-58. INTCPS_ILR13 Register Field Descriptions 201
2-59. INTCPS_ILR14 Register Field Descriptions 202
2-60. INTCPS_ILR15 Register Field Descriptions 203
2-61. INTCPS_ILR16 Register Field Descriptions 204
2-62. INTCPS_ILR17 Register Field Descriptions 205
2-63. INTCPS_ILR18 Register Field Descriptions 206
2-64. INTCPS_ILR19 Register Field Descriptions 207
2-65. INTCPS_ILR20 Register Field Descriptions 208
2-66. INTCPS_ILR21 Register Field Descriptions 209
2-67. INTCPS_ILR22 Register Field Descriptions 210
2-68. INTCPS_ILR23 Register Field Descriptions 211
2-69. INTCPS_ILR24 Register Field Descriptions 212
2-70. INTCPS_ILR25 Register Field Descriptions 213
2-71. INTCPS_ILR26 Register Field Descriptions 214
2-72. INTCPS_ILR27 Register Field Descriptions 215
2-73. INTCPS_ILR28 Register Field Descriptions 216
2-74. INTCPS_ILR29 Register Field Descriptions 217
2-75. INTCPS_ILR30 Register Field Descriptions 218
2-76. INTCPS_ILR31 Register Field Descriptions 219
2-77. INTCPS_ILR32 Register Field Descriptions 220
2-78. INTCPS_ILR33 Register Field Descriptions 221
2-79. INTCPS_ILR34 Register Field Descriptions 222
2-80. INTCPS_ILR35 Register Field Descriptions 223
2-81. INTCPS_ILR36 Register Field Descriptions 224
2-82. INTCPS_ILR37 Register Field Descriptions 225
2-83. INTCPS_ILR38 Register Field Descriptions 226
2-84. INTCPS_ILR39 Register Field Descriptions 227
2-85. INTCPS_ILR40 Register Field Descriptions 228
2-86. INTCPS_ILR41 Register Field Descriptions 229
2-87. INTCPS_ILR42 Register Field Descriptions 230
2-88. INTCPS_ILR43 Register Field Descriptions 231
2-89. INTCPS_ILR44 Register Field Descriptions 232
2-90. INTCPS_ILR45 Register Field Descriptions 233
2-91. INTCPS_ILR46 Register Field Descriptions 234
2-92. INTCPS_ILR47 Register Field Descriptions 235
2-93. INTCPS_ILR48 Register Field Descriptions 236
2-94. INTCPS_ILR49 Register Field Descriptions 237
2-95. INTCPS_ILR50 Register Field Descriptions 238
2-96. INTCPS_ILR51 Register Field Descriptions 239

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2-97. INTCPS_ILR52 Register Field Descriptions 240
2-98. INTCPS_ILR53 Register Field Descriptions 241
2-99. INTCPS_ILR54 Register Field Descriptions 242
2-100. INTCPS_ILR55 Register Field Descriptions 243
2-101. INTCPS_ILR56 Register Field Descriptions 244
2-102. INTCPS_ILR57 Register Field Descriptions 245
2-103. INTCPS_ILR58 Register Field Descriptions 246
2-104. INTCPS_ILR59 Register Field Descriptions 247
2-105. INTCPS_ILR60 Register Field Descriptions 248
2-106. INTCPS_ILR61 Register Field Descriptions 249
2-107. INTCPS_ILR62 Register Field Descriptions 250
2-108. INTCPS_ILR63 Register Field Descriptions 251
2-109. INTCPS_ILR64 Register Field Descriptions 252
2-110. INTCPS_ILR65 Register Field Descriptions 253
2-111. INTCPS_ILR66 Register Field Descriptions 254
2-112. INTCPS_ILR67 Register Field Descriptions 255
2-113. INTCPS_ILR68 Register Field Descriptions 256
2-114. INTCPS_ILR69 Register Field Descriptions 257
2-115. INTCPS_ILR70 Register Field Descriptions 258
2-116. INTCPS_ILR71 Register Field Descriptions 259
2-117. INTCPS_ILR72 Register Field Descriptions 260
2-118. INTCPS_ILR73 Register Field Descriptions 261
2-119. INTCPS_ILR74 Register Field Descriptions 262
2-120. INTCPS_ILR75 Register Field Descriptions 263
2-121. INTCPS_ILR76 Register Field Descriptions 264
2-122. INTCPS_ILR77 Register Field Descriptions 265
2-123. INTCPS_ILR78 Register Field Descriptions 266
2-124. INTCPS_ILR79 Register Field Descriptions 267
2-125. INTCPS_ILR80 Register Field Descriptions 268
2-126. INTCPS_ILR81 Register Field Descriptions 269
2-127. INTCPS_ILR82 Register Field Descriptions 270
2-128. INTCPS_ILR83 Register Field Descriptions 271
2-129. INTCPS_ILR84 Register Field Descriptions 272
2-130. INTCPS_ILR85 Register Field Descriptions 273
2-131. INTCPS_ILR86 Register Field Descriptions 274
2-132. INTCPS_ILR87 Register Field Descriptions 275
2-133. INTCPS_ILR88 Register Field Descriptions 276
2-134. INTCPS_ILR89 Register Field Descriptions 277
2-135. INTCPS_ILR90 Register Field Descriptions 278
2-136. INTCPS_ILR91 Register Field Descriptions 279
2-137. INTCPS_ILR92 Register Field Descriptions 280
2-138. INTCPS_ILR93 Register Field Descriptions 281
2-139. INTCPS_ILR94 Register Field Descriptions 282
2-140. INTCPS_ILR95 Register Field Descriptions 283
2-141. INTCPS_SIR_IRQ Register Field Descriptions 284
2-142. INTCPS_ILR96 Register Field Descriptions 285
2-143. INTCPS_ILR97 Register Field Descriptions 286
2-144. INTCPS_ILR98 Register Field Descriptions 287
2-145. INTCPS_ILR99 Register Field Descriptions 288

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2-146. INTCPS_ILR100 Register Field Descriptions 289
2-147. INTCPS_ILR101 Register Field Descriptions 290
2-148. INTCPS_ILR102 Register Field Descriptions 291
2-149. INTCPS_ILR103 Register Field Descriptions 292
2-150. INTCPS_ILR104 Register Field Descriptions 293
2-151. INTCPS_ILR105 Register Field Descriptions 294
2-152. INTCPS_SIR_FIQ Register Field Descriptions 295
2-153. INTCPS_ILR106 Register Field Descriptions 296
2-154. INTCPS_ILR107 Register Field Descriptions 297
2-155. INTCPS_ILR108 Register Field Descriptions 298
2-156. INTCPS_ILR109 Register Field Descriptions 299
2-157. INTCPS_ILR110 Register Field Descriptions 300
2-158. INTCPS_ILR111 Register Field Descriptions 301
2-159. INTCPS_ILR112 Register Field Descriptions 302
2-160. INTCPS_ILR113 Register Field Descriptions 303
2-161. INTCPS_ILR114 Register Field Descriptions 304
2-162. INTCPS_ILR115 Register Field Descriptions 305
2-163. INTCPS_CONTROL Register Field Descriptions 306
2-164. INTCPS_ILR116 Register Field Descriptions 307
2-165. INTCPS_ILR117 Register Field Descriptions 308
2-166. INTCPS_ILR118 Register Field Descriptions 309
2-167. INTCPS_ILR119 Register Field Descriptions 310
2-168. INTCPS_ILR120 Register Field Descriptions 311
2-169. INTCPS_ILR121 Register Field Descriptions 312
2-170. INTCPS_ILR122 Register Field Descriptions 313
2-171. INTCPS_ILR123 Register Field Descriptions 314
2-172. INTCPS_ILR124 Register Field Descriptions 315
2-173. INTCPS_ILR125 Register Field Descriptions 316
2-174. INTCPS_PROTECTION Register Field Descriptions 317
2-175. INTCPS_ILR126 Register Field Descriptions 318
2-176. INTCPS_ILR127 Register Field Descriptions 319
2-177. INTCPS_IDLE Register Field Descriptions 320
2-178. INTCPS_IRQ_PRIORITY Register Field Descriptions 321
3-1. L3 Memory Map 323
3-2. M3 Processor Memory Map 326
3-3. L4 Fast Peripheral Memory Map 328
3-4. L4_PER Peripheral Memory Map 329
3-5. L4_FW Firewall Configuration Memory Map 332
3-6. L4_WKUP Peripheral Memory Map 334
4-1. PRUSS_INTC REGISTERS 339
4-2. REVID Register Field Descriptions 342
4-3. CMR0 Register Field Descriptions 343
4-4. CMR1 Register Field Descriptions 344
4-5. CMR2 Register Field Descriptions 345
4-6. CMR3 Register Field Descriptions 346
4-7. CMR4 Register Field Descriptions 347
4-8. CMR5 Register Field Descriptions 348
4-9. CMR6 Register Field Descriptions 349
4-10. CMR7 Register Field Descriptions 350

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4-11. CMR8 Register Field Descriptions 351
4-12. CMR9 Register Field Descriptions 352
4-13. CMR10 Register Field Descriptions 353
4-14. CMR11 Register Field Descriptions 354
4-15. CMR12 Register Field Descriptions 355
4-16. CMR13 Register Field Descriptions 356
4-17. CMR14 Register Field Descriptions 357
4-18. CMR15 Register Field Descriptions 358
4-19. GPIR Register Field Descriptions 359
4-20. GER Register Field Descriptions 360
4-21. HMR0 Register Field Descriptions 361
4-22. HMR1 Register Field Descriptions 362
4-23. HMR2 Register Field Descriptions 363
4-24. HIPIR0 Register Field Descriptions 364
4-25. HIPIR1 Register Field Descriptions 365
4-26. HIPIR2 Register Field Descriptions 366
4-27. HIPIR3 Register Field Descriptions 367
4-28. HIPIR4 Register Field Descriptions 368
4-29. HIPIR5 Register Field Descriptions 369
4-30. HIPIR6 Register Field Descriptions 370
4-31. HIPIR7 Register Field Descriptions 371
4-32. HIPIR8 Register Field Descriptions 372
4-33. HIPIR9 Register Field Descriptions 373
4-34. GNLR Register Field Descriptions 374
4-35. SISR Register Field Descriptions 375
4-36. SIPR0 Register Field Descriptions 376
4-37. SIPR1 Register Field Descriptions 377
4-38. SITR0 Register Field Descriptions 378
4-39. SITR1 Register Field Descriptions 379
4-40. SICR Register Field Descriptions 380
4-41. CR Register Field Descriptions 381
4-42. EISR Register Field Descriptions 382
4-43. HINLR0 Register Field Descriptions 383
4-44. HINLR1 Register Field Descriptions 384
4-45. HINLR2 Register Field Descriptions 385
4-46. HINLR3 Register Field Descriptions 386
4-47. HINLR4 Register Field Descriptions 387
4-48. HINLR5 Register Field Descriptions 388
4-49. HINLR6 Register Field Descriptions 389
4-50. HINLR7 Register Field Descriptions 390
4-51. HINLR8 Register Field Descriptions 391
4-52. HINLR9 Register Field Descriptions 392
4-53. EICR Register Field Descriptions 393
4-54. SRSR0 Register Field Descriptions 394
4-55. SRSR1 Register Field Descriptions 395
4-56. HIEISR Register Field Descriptions 396
4-57. HIER Register Field Descriptions 397
4-58. HIDISR Register Field Descriptions 398
4-59. SECR0 Register Field Descriptions 399

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4-60. SECR1 Register Field Descriptions 400
4-61. ESR0 Register Field Descriptions 401
4-62. ERS1 Register Field Descriptions 402
4-63. ECR0 Register Field Descriptions 403
4-64. ECR1 Register Field Descriptions 404
4-65. PRUSS_PRU_CTRL REGISTERS 404
4-66. CONTROL Register Field Descriptions 405
4-67. CYCLE Register Field Descriptions 406
4-68. STALL Register Field Descriptions 407
4-69. CTBIR0 Register Field Descriptions 408
4-70. CTBIR1 Register Field Descriptions 409
4-71. STATUS Register Field Descriptions 410
4-72. CTPPR0 Register Field Descriptions 411
4-73. CTPPR1 Register Field Descriptions 412
4-74. WAKEUP_EN Register Field Descriptions 413
4-75. PRUSS_PRU_DEBUG REGISTERS 413
4-76. GPR0 Register Field Descriptions 415
4-77. GPR25 Register Field Descriptions 416
4-78. GPR26 Register Field Descriptions 417
4-79. GPR27 Register Field Descriptions 418
4-80. GPR28 Register Field Descriptions 419
4-81. GPR29 Register Field Descriptions 420
4-82. GPR3 Register Field Descriptions 421
4-83. GPR30 Register Field Descriptions 422
4-84. GPR31 Register Field Descriptions 423
4-85. CTR0 Register Field Descriptions 424
4-86. CTR1 Register Field Descriptions 425
4-87. CTR2 Register Field Descriptions 426
4-88. CTR3 Register Field Descriptions 427
4-89. CTR4 Register Field Descriptions 428
4-90. CTR5 Register Field Descriptions 429
4-91. CTR6 Register Field Descriptions 430
4-92. CTR7 Register Field Descriptions 431
4-93. GPR4 Register Field Descriptions 432
4-94. CTR8 Register Field Descriptions 433
4-95. CTR9 Register Field Descriptions 434
4-96. CTR10 Register Field Descriptions 435
4-97. CTR11 Register Field Descriptions 436
4-98. CTR12 Register Field Descriptions 437
4-99. CTR13 Register Field Descriptions 438
4-100. CTR14 Register Field Descriptions 439
4-101. CTR15 Register Field Descriptions 440
4-102. CTR16 Register Field Descriptions 441
4-103. CTR17 Register Field Descriptions 442
4-104. GPR5 Register Field Descriptions 443
4-105. CTR18 Register Field Descriptions 444
4-106. CTR19 Register Field Descriptions 445
4-107. CTR20 Register Field Descriptions 446
4-108. CTR21 Register Field Descriptions 447

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4-109. CTR22 Register Field Descriptions 448
4-110. CTR23 Register Field Descriptions 449
4-111. CTR24 Register Field Descriptions 450
4-112. GPR6 Register Field Descriptions 451
4-113. GPR7 Register Field Descriptions 452
4-114. GPR8 Register Field Descriptions 453
4-115. GPR9 Register Field Descriptions 454
4-116. GPR1 Register Field Descriptions 455
4-117. GPR10 Register Field Descriptions 456
4-118. GPR11 Register Field Descriptions 457
4-119. GPR12 Register Field Descriptions 458
4-120. GPR13 Register Field Descriptions 459
4-121. GPR14 Register Field Descriptions 460
4-122. GPR15 Register Field Descriptions 461
4-123. GPR16 Register Field Descriptions 462
4-124. GPR17 Register Field Descriptions 463
4-125. GPR18 Register Field Descriptions 464
4-126. GPR19 Register Field Descriptions 465
4-127. GPR2 Register Field Descriptions 466
4-128. GPR20 Register Field Descriptions 467
4-129. GPR21 Register Field Descriptions 468
4-130. GPR22 Register Field Descriptions 469
4-131. GPR23 Register Field Descriptions 470
4-132. GPR24 Register Field Descriptions 471
4-133. PRUSS_CFG REGISTERS 471
4-134. REVID Register Field Descriptions 472
4-135. GPCFG1 Register Field Descriptions 473
4-136. CGR Register Field Descriptions 474
4-137. ISRP Register Field Descriptions 475
4-138. ISP Register Field Descriptions 476
4-139. IESP Register Field Descriptions 477
4-140. IECP Register Field Descriptions 478
4-141. SCRP Register Field Descriptions 479
4-142. SYSCFG Register Field Descriptions 480
4-143. PMAO Register Field Descriptions 481
4-144. MII_RT Register Field Descriptions 482
4-145. IEPCLK Register Field Descriptions 483
4-146. SPP Register Field Descriptions 484
4-147. PIN_MX Register Field Descriptions 485
4-148. GPCFG0 Register Field Descriptions 486
4-149. PRUSS_UART REGISTERS 486
4-150. UART RBR/TBR Registers Register Field Descriptions 488
4-151. UART Line Control Register Register Field Descriptions 489
4-152. UART Modem Control Register Register Field Descriptions 490
4-153. UART Line Status Register Register Field Descriptions 491
4-154. UART Modem Status Register Register Field Descriptions 492
4-155. UART Scratch Register Register Field Descriptions 493
4-156. UART Divisor Register (LSB) Register Field Descriptions 494
4-157. UART Divisor Register (MSB) Register Field Descriptions 495

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4-158. UART Interrupt Enable Register Register Field Descriptions 496
4-159. UART Peripheral ID Register Register Field Descriptions 497
4-160. UART PowerManagement and Emulation Register Register Field Descriptions 498
4-161. UART Mode Definition Register Register Field Descriptions 499
4-162. UART Interrupt Identification Register / FIFO Control Register Register Field Descriptions 500
4-163. PRUSS_IEP REGISTERS 500
4-164. GLOBAL_CFG Register Field Descriptions 502
4-165. CMP7 Register Field Descriptions 503
4-166. COUNT Register Field Descriptions 504
4-167. RXIPG0 Register Field Descriptions 505
4-168. RXIPG1 Register Field Descriptions 506
4-169. CAP_CFG Register Field Descriptions 507
4-170. CAP_STATUS Register Field Descriptions 508
4-171. CAPR0 Register Field Descriptions 509
4-172. SYNC_CTRL Register Field Descriptions 510
4-173. SYNC_FIRST_STAT Register Field Descriptions 511
4-174. SYNC0_STAT Register Field Descriptions 512
4-175. SYNC1_STAT Register Field Descriptions 513
4-176. SYNC_PWIDTH Register Field Descriptions 514
4-177. SYNC0_PERIOD Register Field Descriptions 515
4-178. CAPR1 Register Field Descriptions 516
4-179. SYNC1_DELAY Register Field Descriptions 517
4-180. SYNC_START Register Field Descriptions 518
4-181. CAPR2 Register Field Descriptions 519
4-182. CAPR3 Register Field Descriptions 520
4-183. GLOBAL_STATUS Register Field Descriptions 521
4-184. CAPR4 Register Field Descriptions 522
4-185. CAPR5 Register Field Descriptions 523
4-186. CAPR6 Register Field Descriptions 524
4-187. WD_PREDIV Register Field Descriptions 525
4-188. PDI_WD_TIM Register Field Descriptions 526
4-189. CAPF6 Register Field Descriptions 527
4-190. PD_WD_TIM Register Field Descriptions 528
4-191. WD_STATUS Register Field Descriptions 529
4-192. WD_EXP_CNT Register Field Descriptions 530
4-193. WD_CTRL Register Field Descriptions 531
4-194. CAPR7 Register Field Descriptions 532
4-195. CAPF7 Register Field Descriptions 533
4-196. CMP_CFG Register Field Descriptions 534
4-197. CMP_STATUS Register Field Descriptions 535
4-198. CMP0 Register Field Descriptions 536
4-199. CMP1 Register Field Descriptions 537
4-200. DIGIO_CTRL Register Field Descriptions 538
4-201. DIGIO_STATUS Register Field Descriptions 539
4-202. DIGIO_DATA_IN Register Field Descriptions 540
4-203. DIGIO_DATA_IN_RAW Register Field Descriptions 541
4-204. DIGIO_DATA_OUT Register Field Descriptions 542
4-205. DIGIO_DATA_OUT_EN Register Field Descriptions 543
4-206. DIGIO_EXP Register Field Descriptions 544

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4-207. COMPEN Register Field Descriptions 545
4-208. CMP2 Register Field Descriptions 546
4-209. CMP3 Register Field Descriptions 547
4-210. CMP4 Register Field Descriptions 548
4-211. CMP5 Register Field Descriptions 549
4-212. CMP6 Register Field Descriptions 550
4-213. PRUSS_ECAP REGISTERS 550
4-214. TSCNT Register Field Descriptions 551
4-215. CAP2 Register Field Descriptions 552
4-216. CAP3 Register Field Descriptions 553
4-217. CAP4 Register Field Descriptions 554
4-218. ECCTL1 Register Field Descriptions 555
4-219. CNTPHS Register Field Descriptions 557
4-220. ECCTL2 Register Field Descriptions 558
4-221. ECEINT Register Field Descriptions 560
4-222. ECFLG Register Field Descriptions 561
4-223. ECCLR Register Field Descriptions 562
4-224. ECFRC Register Field Descriptions 563
4-225. CAP1 Register Field Descriptions 564
4-226. PID Register Field Descriptions 565
4-227. PRUSS_MII_RT_CFG REGISTERS 565
4-228. RXCFG0 Register Field Descriptions 566
4-229. TXCFG0 Register Field Descriptions 567
4-230. TXCFG1 Register Field Descriptions 568
4-231. TX_CRC0 Register Field Descriptions 569
4-232. TX_CRC1 Register Field Descriptions 570
4-233. RXCFG1 Register Field Descriptions 571
4-234. TX_IPG0 Register Field Descriptions 572
4-235. TX_IPG1 Register Field Descriptions 573
4-236. PRS0 Register Field Descriptions 574
4-237. PRS1 Register Field Descriptions 575
4-238. RX_FRMS0 Register Field Descriptions 576
4-239. RX_FRMS1 Register Field Descriptions 577
4-240. RX_PCNT0 Register Field Descriptions 578
4-241. RX_PCNT1 Register Field Descriptions 579
4-242. RX_ERR0 Register Field Descriptions 580
4-243. RX_ERR1 Register Field Descriptions 581
4-244. PRUSS_MII_MDIO REGISTERS 581
4-245. VER Register Field Descriptions 583
4-246. LINK Register Field Descriptions 584
4-247. USERACCESS0 Register Field Descriptions 585
4-248. USERPHYSEL0 Register Field Descriptions 586
4-249. USERACCESS1 Register Field Descriptions 587
4-250. USERPHYSEL1 Register Field Descriptions 588
4-251. LINKINTRAW Register Field Descriptions 589
4-252. LINKINTMASKED Register Field Descriptions 590
4-253. USERINTRAW Register Field Descriptions 591
4-254. USERINTMASKED Register Field Descriptions 592
4-255. CONTROL Register Field Descriptions 593

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4-256. USERINTMASKSET Register Field Descriptions 594
4-257. USERINTMASKCLR Register Field Descriptions 595
4-258. ALIVE Register Field Descriptions 596
5-1. SGX530_MMAV10GS70 REGISTERS 599
5-2. EUR_CR_CLKGATECTL Register Field Descriptions 609
5-3. EUR_CR_PERF_COUNTER8 Register Field Descriptions 611
5-4. EUR_CR_PIXELBE Register Field Descriptions 612
5-5. EUR_CR_ISP_RENDER Register Field Descriptions 613
5-6. EUR_CR_ISP_RGN_BASE Register Field Descriptions 614
5-7. EUR_CR_ISP_RENDBOX1 Register Field Descriptions 615
5-8. EUR_CR_ISP_RENDBOX2 Register Field Descriptions 616
5-9. EUR_CR_ISP_IPFMISC Register Field Descriptions 617
5-10. EUR_CR_ISP_OGL_MODE Register Field Descriptions 618
5-11. EUR_CR_ISP_PERPENDICULAR Register Field Descriptions 619
5-12. EUR_CR_ISP_CULLVALUE Register Field Descriptions 620
5-13. EUR_CR_ISP_DBIAS Register Field Descriptions 621
5-14. EUR_CR_ISP_START_RENDER Register Field Descriptions 622
5-15. EUR_CR_3D_AA_MODE Register Field Descriptions 623
5-16. EUR_CR_ISP_BREAK Register Field Descriptions 624
5-17. EUR_CR_ISP_3DCONTEXT Register Field Descriptions 625
5-18. EUR_CR_ISP_FPU Register Field Descriptions 626
5-19. EUR_CR_TSP_PARAMETER_CACHE Register Field Descriptions 627
5-20. EUR_CR_TSP_PARAMETER_FETCH Register Field Descriptions 628
5-21. EUR_CR_ISP_ZLSCTL Register Field Descriptions 629
5-22. EUR_CR_ISP_ZLOAD_BASE Register Field Descriptions 630
5-23. EUR_CR_ISP_ZSTORE_BASE Register Field Descriptions 631
5-24. EUR_CR_ISP_STENCIL_LOAD_BASE Register Field Descriptions 632
5-25. EUR_CR_ISP_STENCIL_STORE_BASE Register Field Descriptions 633
5-26. EUR_CR_ISP_VISTEST_CTRL Register Field Descriptions 634
5-27. EUR_CR_ISP_VISTEST_VISIBLE0 Register Field Descriptions 635
5-28. EUR_CR_ISP_VISTEST_VISIBLE1 Register Field Descriptions 636
5-29. EUR_CR_ISP_VISTEST_VISIBLE2 Register Field Descriptions 637
5-30. EUR_CR_ISP_VISTEST_VISIBLE3 Register Field Descriptions 638
5-31. EUR_CR_ISP_VISTEST_VISIBLE4 Register Field Descriptions 639
5-32. EUR_CR_ISP_VISTEST_VISIBLE5 Register Field Descriptions 640
5-33. EUR_CR_ISP_VISTEST_VISIBLE6 Register Field Descriptions 641
5-34. EUR_CR_ISP_VISTEST_VISIBLE7 Register Field Descriptions 642
5-35. EUR_CR_ISP_BGOBJDEPTH Register Field Descriptions 643
5-36. EUR_CR_ISP_BGOBJ Register Field Descriptions 644
5-37. EUR_CR_ISP_BGOBJTAG Register Field Descriptions 645
5-38. EUR_CR_ISP_MULTISAMPLECTL Register Field Descriptions 646
5-39. EUR_CR_ISP_SIG1 Register Field Descriptions 647
5-40. EUR_CR_ISP_SIG2 Register Field Descriptions 648
5-41. EUR_CR_ISP_SIG3 Register Field Descriptions 649
5-42. EUR_CR_ISP_SIG4 Register Field Descriptions 650
5-43. EUR_CR_ISP_TAGCTRL Register Field Descriptions 651
5-44. EUR_CR_ISP_STATUS1 Register Field Descriptions 652
5-45. EUR_CR_ISP_STATUS2 Register Field Descriptions 653
5-46. EUR_CR_ISP_DEPTHSORT Register Field Descriptions 654

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5-47. EUR_CR_PIXELBE_EMIT Register Field Descriptions 655
5-48. EUR_CR_SOFT_RESET Register Field Descriptions 656
5-49. EUR_CR_DPM_3D_PAGE_TABLE_BASE Register Field Descriptions 657
5-50. EUR_CR_DPM_3D_FREE_LIST Register Field Descriptions 658
5-51. EUR_CR_DPM_PDS_PAGE_THRESHOLD Register Field Descriptions 659
5-52. EUR_CR_DPM_TA_ALLOC_PAGE_TABLE_BASE Register Field Descriptions 660
5-53. EUR_CR_DPM_TA_ALLOC_FREE_LIST Register Field Descriptions 661
5-54. EUR_CR_DPM_TA_PAGE_THRESHOLD Register Field Descriptions 662
5-55. EUR_CR_DPM_ZLS_PAGE_THRESHOLD Register Field Descriptions 663
5-56. EUR_CR_DPM_TA_GLOBAL_LIST Register Field Descriptions 664
5-57. EUR_CR_DPM_STATE_TABLE_BASE Register Field Descriptions 665
5-58. EUR_CR_DPM_STATE_CONTEXT_ID Register Field Descriptions 666
5-59. EUR_CR_DPM_CONTROL_TABLE_BASE Register Field Descriptions 667
5-60. EUR_CR_DPM_START_OF_CONTEXT_PAGES_ALLOCATED Register Field Descriptions 668
5-61. EUR_CR_DPM_3D_DEALLOCATE Register Field Descriptions 669
5-62. EUR_CR_CORE_ID Register Field Descriptions 670
5-63. EUR_CR_DPM_ALLOC Register Field Descriptions 671
5-64. EUR_CR_DPM_DALLOC Register Field Descriptions 672
5-65. EUR_CR_DPM_TA_ALLOC Register Field Descriptions 673
5-66. EUR_CR_DPM_3D Register Field Descriptions 674
5-67. EUR_CR_DPM_PARTIAL_RENDER Register Field Descriptions 675
5-68. EUR_CR_DPM_LSS_PARTIAL_CONTEXT Register Field Descriptions 676
5-69. EUR_CR_DPM_START_OF_3D_CONTEXT_PAGES_ALLOCATED Register Field Descriptions 677
5-70. EUR_CR_DPM_TASK_3D_FREE Register Field Descriptions 678
5-71. EUR_CR_DPM_TASK_TA_FREE Register Field Descriptions 679
5-72. EUR_CR_DPM_TASK_STATE Register Field Descriptions 680
5-73. EUR_CR_DPM_TASK_CONTROL Register Field Descriptions 681
5-74. EUR_CR_DPM_OUTOFMEM Register Field Descriptions 682
5-75. EUR_CR_DPM_FREE_CONTEXT Register Field Descriptions 683
5-76. EUR_CR_DPM_3D_TIMEOUT Register Field Descriptions 684
5-77. EUR_CR_DPM_TA_EVM Register Field Descriptions 685
5-78. EUR_CR_DPM_TA_ALLOC_FREE_LIST_STATUS1 Register Field Descriptions 686
5-79. EUR_CR_DPM_3D_FREE_LIST_STATUS1 Register Field Descriptions 687
5-80. EUR_CR_DPM_TA_ALLOC_FREE_LIST_STATUS2 Register Field Descriptions 688
5-81. EUR_CR_DPM_3D_FREE_LIST_STATUS2 Register Field Descriptions 689
5-82. EUR_CR_DPM_ABORT_STATUS_MTILE Register Field Descriptions 690
5-83. EUR_CR_DPM_PAGE_STATUS Register Field Descriptions 691
5-84. EUR_CR_DPM_PAGE Register Field Descriptions 692
5-85. EUR_CR_DPM_GLOBAL_PAGE_STATUS Register Field Descriptions 693
5-86. EUR_CR_DPM_GLOBAL_PAGE Register Field Descriptions 694
5-87. EUR_CR_DPM_REQUESTING Register Field Descriptions 695
5-88. EUR_CR_DPM_RNG_HDR_PARSER Register Field Descriptions 696
5-89. EUR_CR_CORE_REVISION Register Field Descriptions 697
5-90. EUR_CR_TSP_CONFIG Register Field Descriptions 698
5-91. EUR_CR_CACHE_CTRL Register Field Descriptions 699
5-92. EUR_CR_DESIGNER_REV_FIELD1 Register Field Descriptions 700
5-93. EUR_CR_TRIGGER Register Field Descriptions 701
5-94. EUR_CR_USE_CTRL Register Field Descriptions 702
5-95. EUR_CR_USE_LD_REDIRECT Register Field Descriptions 703

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5-96. EUR_CR_USE_CACHE Register Field Descriptions 704
5-97. EUR_CR_USE_CODE_BASE00 Register Field Descriptions 705
5-98. EUR_CR_USE_CODE_BASE01 Register Field Descriptions 706
5-99. EUR_CR_USE_CODE_BASE02 Register Field Descriptions 707
5-100. EUR_CR_USE_CODE_BASE03 Register Field Descriptions 708
5-101. EUR_CR_USE_CODE_BASE04 Register Field Descriptions 709
5-102. EUR_CR_USE_CODE_BASE05 Register Field Descriptions 710
5-103. EUR_CR_USE_CODE_BASE06 Register Field Descriptions 711
5-104. EUR_CR_USE_CODE_BASE07 Register Field Descriptions 712
5-105. EUR_CR_USE_CODE_BASE08 Register Field Descriptions 713
5-106. EUR_CR_USE_CODE_BASE09 Register Field Descriptions 714
5-107. EUR_CR_USE_CODE_BASE10 Register Field Descriptions 715
5-108. EUR_CR_USE_CODE_BASE11 Register Field Descriptions 716
5-109. EUR_CR_USE_CODE_BASE12 Register Field Descriptions 717
5-110. EUR_CR_USE_CODE_BASE13 Register Field Descriptions 718
5-111. EUR_CR_USE_CODE_BASE14 Register Field Descriptions 719
5-112. EUR_CR_USE_CODE_BASE15 Register Field Descriptions 720
5-113. EUR_CR_EVENT_PDS_ENABLE2 Register Field Descriptions 721
5-114. EUR_CR_PDS_CACHE_ENABLE Register Field Descriptions 722
5-115. EUR_CR_EVENT_PDS_ENABLE Register Field Descriptions 723
5-116. EUR_CR_EVENT_PIXEL_PDS_EXEC Register Field Descriptions 725
5-117. EUR_CR_EVENT_PIXEL_PDS_DATA Register Field Descriptions 726
5-118. EUR_CR_EVENT_PIXEL_PDS_INFO Register Field Descriptions 727
5-119. EUR_CR_EVENT_OTHER_PDS_EXEC Register Field Descriptions 728
5-120. EUR_CR_EVENT_OTHER_PDS_DATA Register Field Descriptions 729
5-121. EUR_CR_EVENT_OTHER_PDS_INFO Register Field Descriptions 730
5-122. EUR_CR_DMS_CTRL Register Field Descriptions 731
5-123. EUR_CR_DMS_AGE Register Field Descriptions 732
5-124. EUR_CR_USE_G0 Register Field Descriptions 733
5-125. EUR_CR_USE_G1 Register Field Descriptions 734
5-126. EUR_CR_USE_FILTER0_LEFT Register Field Descriptions 735
5-127. EUR_CR_USE_FILTER0_RIGHT Register Field Descriptions 736
5-128. EUR_CR_USE_FILTER0_EXTRA Register Field Descriptions 737
5-129. EUR_CR_USE_FILTER1_LEFT Register Field Descriptions 738
5-130. EUR_CR_USE_FILTER1_RIGHT Register Field Descriptions 739
5-131. EUR_CR_USE_FILTER1_EXTRA Register Field Descriptions 740
5-132. EUR_CR_USE_FILTER_TABLE Register Field Descriptions 741
5-133. EUR_CR_EVENT_HOST_ENABLE2 Register Field Descriptions 742
5-134. EUR_CR_USE_ST_RANGE Register Field Descriptions 743
5-135. EUR_CR_USE0_DM_SLOT Register Field Descriptions 744
5-136. EUR_CR_USE1_DM_SLOT Register Field Descriptions 746
5-137. EUR_CR_USE_EXEF Register Field Descriptions 748
5-138. EUR_CR_PDS_EXEC_BASE Register Field Descriptions 749
5-139. EUR_CR_PDS Register Field Descriptions 750
5-140. EUR_CR_EVENT_KICKER Register Field Descriptions 751
5-141. EUR_CR_EVENT_HOST_CLEAR2 Register Field Descriptions 752
5-142. EUR_CR_EVENT_KICK Register Field Descriptions 753
5-143. EUR_CR_EVENT_TIMER Register Field Descriptions 754
5-144. EUR_CR_PDS_INV0 Register Field Descriptions 755

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5-145. EUR_CR_PDS_INV1 Register Field Descriptions 756
5-146. EUR_CR_PDS_INV2 Register Field Descriptions 757
5-147. EUR_CR_PDS_INV3 Register Field Descriptions 758
5-148. EUR_CR_PDS_INV_CSC Register Field Descriptions 759
5-149. EUR_CR_USE_GLOBCOM0 Register Field Descriptions 760
5-150. EUR_CR_USE_GLOBCOM1 Register Field Descriptions 761
5-151. EUR_CR_USE_GLOBCOM2 Register Field Descriptions 762
5-152. EUR_CR_DESIGNER_REV_FIELD2 Register Field Descriptions 763
5-153. EUR_CR_EVENT_STATUS2 Register Field Descriptions 764
5-154. EUR_CR_USE_GLOBCOM3 Register Field Descriptions 765
5-155. EUR_CR_USE_GLOBCOM4 Register Field Descriptions 766
5-156. EUR_CR_USE_GLOBCOM5 Register Field Descriptions 767
5-157. EUR_CR_USE_GLOBCOM6 Register Field Descriptions 768
5-158. EUR_CR_USE_GLOBCOM7 Register Field Descriptions 769
5-159. EUR_CR_LOOPBACK Register Field Descriptions 770
5-160. EUR_CR_USE0_SERV_PIXEL Register Field Descriptions 771
5-161. EUR_CR_USE0_SERV_VERTEX Register Field Descriptions 772
5-162. EUR_CR_USE0_SERV_EVENT Register Field Descriptions 773
5-163. EUR_CR_USE1_SERV_PIXEL Register Field Descriptions 774
5-164. EUR_CR_USE1_SERV_VERTEX Register Field Descriptions 775
5-165. EUR_CR_USE1_SERV_EVENT Register Field Descriptions 776
5-166. EUR_CR_USE_FILTER2_LEFT Register Field Descriptions 777
5-167. EUR_CR_USE_FILTER2_RIGHT Register Field Descriptions 778
5-168. EUR_CR_USE_FILTER2_EXTRA Register Field Descriptions 779
5-169. EUR_CR_PDS_PC_BASE Register Field Descriptions 780
5-170. EUR_CR_USE_MOE Register Field Descriptions 781
5-171. EUR_CR_EVENT_STATUS Register Field Descriptions 782
5-172. EUR_CR_EVENT_HOST_ENABLE Register Field Descriptions 784
5-173. EUR_CR_BIF_CTRL Register Field Descriptions 786
5-174. EUR_CR_BIF_INT_STAT Register Field Descriptions 787
5-175. EUR_CR_EVENT_HOST_CLEAR Register Field Descriptions 788
5-176. EUR_CR_BIF_FAULT Register Field Descriptions 789
5-177. EUR_CR_PDS_CACHE_STATUS Register Field Descriptions 790
5-178. EUR_CR_PDS_CACHE_HOST_ENABLE Register Field Descriptions 791
5-179. EUR_CR_PDS_CACHE_HOST_CLEAR Register Field Descriptions 792
5-180. EUR_CR_BIF_DIR_LIST_BASE0 Register Field Descriptions 793
5-181. EUR_CR_BIF_TA_REQ_BASE Register Field Descriptions 794
5-182. EUR_CR_BIF_MEM_REQ_STAT Register Field Descriptions 795
5-183. EUR_CR_BIF_3D_REQ_BASE Register Field Descriptions 796
5-184. EUR_CR_BIF_ZLS_REQ_BASE Register Field Descriptions 797
5-185. EUR_CR_CLKGATESTATUS Register Field Descriptions 798
5-186. EUR_CR_VDM_START Register Field Descriptions 799
5-187. EUR_CR_TE_AA Register Field Descriptions 800
5-188. EUR_CR_TE_MTILE1 Register Field Descriptions 801
5-189. EUR_CR_TE_MTILE2 Register Field Descriptions 802
5-190. EUR_CR_TE_SCREEN Register Field Descriptions 803
5-191. EUR_CR_TE_MTILE Register Field Descriptions 804
5-192. EUR_CR_TE_PSG Register Field Descriptions 805
5-193. EUR_CR_TE_PSGREGION_BASE Register Field Descriptions 806

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5-194. EUR_CR_TE_TPC_BASE Register Field Descriptions 807
5-195. EUR_CR_TE_TPCCONTROL Register Field Descriptions 808
5-196. EUR_CR_TE_RGNBBOX_X Register Field Descriptions 809
5-197. EUR_CR_TE_RGNBBOX_Y Register Field Descriptions 810
5-198. EUR_CR_MTE_OTPM_CSM_FLUSH_BASE Register Field Descriptions 811
5-199. EUR_CR_MTE_OTPM_CSM_LOAD_BASE Register Field Descriptions 812
5-200. EUR_CR_VDM_CTRL_STREAM_BASE Register Field Descriptions 813
5-201. EUR_CR_MTE_CTRL Register Field Descriptions 814
5-202. EUR_CR_MTE_WCOMPARE Register Field Descriptions 815
5-203. EUR_CR_MTE_WCLAMP Register Field Descriptions 816
5-204. EUR_CR_MTE_SCREEN Register Field Descriptions 817
5-205. EUR_CR_MTE_OTPM_OP Register Field Descriptions 818
5-206. EUR_CR_MTE_MULTISAMPLECTL Register Field Descriptions 819
5-207. EUR_CR_MTE_SIG1 Register Field Descriptions 820
5-208. EUR_CR_MTE_SIG2 Register Field Descriptions 821
5-209. EUR_CR_TE_STATE Register Field Descriptions 822
5-210. EUR_CR_PERF Register Field Descriptions 823
5-211. EUR_CR_PERF_COUNTER0 Register Field Descriptions 824
5-212. EUR_CR_PERF_COUNTER1 Register Field Descriptions 825
5-213. EUR_CR_PERF_COUNTER2 Register Field Descriptions 826
5-214. EUR_CR_CLIP_SIG1 Register Field Descriptions 827
5-215. EUR_CR_PIXELBE_SIG01 Register Field Descriptions 828
5-216. EUR_CR_PIXELBE_SIG02 Register Field Descriptions 829
5-217. EUR_CR_CLKGATECTLOVR Register Field Descriptions 830
5-218. EUR_CR_PERF_COUNTER3 Register Field Descriptions 831
5-219. EUR_CR_VDM_CONTEXT_STORE_START Register Field Descriptions 832
5-220. EUR_CR_VDM_CONTEXT_STORE_STREAM Register Field Descriptions 833
5-221. EUR_CR_VDM_CONTEXT_STORE_INDEX Register Field Descriptions 834
5-222. EUR_CR_VDM_CONTEXT_STORE_STATUS Register Field Descriptions 835
5-223. EUR_CR_VDM_CONTEXT_STORE_STATE0 Register Field Descriptions 836
5-224. EUR_CR_VDM_CONTEXT_STORE_STATE1 Register Field Descriptions 837
5-225. EUR_CR_PERF_COUNTER4 Register Field Descriptions 838
5-226. EUR_CR_VDM_WAIT_FOR_KICK Register Field Descriptions 839
5-227. EUR_CR_VDM_CONTEXT_STORE_WRAP_INDEX Register Field Descriptions 840
5-228. EUR_CR_VDM_CONTEXT_STORE_WRAPPED Register Field Descriptions 841
5-229. EUR_CR_VDM_CONTEXT_STORE_FAN Register Field Descriptions 842
5-230. EUR_CR_VDM_CONTEXT_STORE_WRAP Register Field Descriptions 843
5-231. EUR_CR_PERF_COUNTER5 Register Field Descriptions 844
5-232. EUR_CR_PERF_COUNTER6 Register Field Descriptions 845
5-233. EUR_CR_PERF_COUNTER7 Register Field Descriptions 846
6-1. Unsupported GPMC Features 850
6-2. GPMC Connectivity Attributes 852
6-3. GPMC Clock Signals 852
6-4. GPMC Pin List 852
6-5. GPMC I/O Description 854
6-6. GPMC Pin Multiplexing Options 854
6-7. GPMC Integration Attributes 859
6-8. GPMC Clocks and Resets 859
6-9. GPMC Hardware Requests 859

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6-10. GPMC Clocks 861
6-11. GPMC_CONFIG1_i Configuration 861
6-12. GPMC Local Power Management Features 861
6-13. GPMC Interrupt Events 862
6-14. Idle Cycle Insertion Configuration 873
6-15. Chip-Select Configuration for NAND Interfacing 902
6-16. ECC Enable Settings 911
6-17. Flattened BCH Codeword Mapping (512 Bytes + 104 Bits) 916
6-18. Aligned Message Byte Mapping in 8-bit NAND 916
6-19. Aligned Message Byte Mapping in 16-bit NAND 917
6-20. Aligned Nibble Mapping of Message in 8-bit NAND 917
6-21. Misaligned Nibble Mapping of Message in 8-bit NAND 917
6-22. Aligned Nibble Mapping of Message in 16-bit NAND 917
6-23. Misaligned Nibble Mapping of Message in 16-bit NAND (1 Unused Nibble) 918
6-24. Misaligned Nibble Mapping of Message in 16-bit NAND (2 Unused Nibble) 918
6-25. Misaligned Nibble Mapping of Message in 16-bit NAND (3 Unused Nibble) 918
6-26. Prefetch Mode Configuration 929
6-27. Write-Posting Mode Configuration 931
6-28. GPMC Configuration in NOR Mode 937
6-29. GPMC Configuration in NAND Mode 937
6-30. Reset GPMC 937
6-31. NOR Memory Type 938
6-32. NOR Chip-Select Configuration 938
6-33. NOR Timings Configuration 938
6-34. WAIT Pin Configuration 938
6-35. Enable Chip-Select 939
6-36. NAND Memory Type 939
6-37. NAND Chip-Select Configuration 939
6-38. Asynchronous Read and Write Operations 939
6-39. ECC Engine 939
6-40. Prefetch and Write-Posting Engine 941
6-41. WAIT Pin Configuration 941
6-42. Enable Chip-Select 941
6-43. Mode Parameters Check List Table 942
6-44. Access Type Parameters Check List Table 942
6-45. Timing Parameters 944
6-46. NAND Formulas Description Table 946
6-47. Synchronous NOR Formulas Description Table 947
6-48. Asynchronous NOR Formulas Description Table 953
6-49. GPMC Signals 955
6-50. Useful Timing Parameters on the Memory Side 957
6-51. Calculating GPMC Timing Parameters 958
6-52. AC Characteristics for Asynchronous Read Access 959
6-53. GPMC Timing Parameters for Asynchronous Read Access 960
6-54. AC Characteristics for Asynchronous Single Write (Memory Side) 961
6-55. GPMC Timing Parameters for Asynchronous Single Write 962
6-56. Supported Memory Interfaces 963
6-57. NAND Interface Bus Operations Summary 965
6-58. NOR Interface Bus Operations Summary 965

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6-59. GPMC Registers 967
6-60. GPMC_REVISION Field Descriptions 968
6-61. GPMC_SYSCONFIG Field Descriptions 968
6-62. GPMC_SYSSTATUS Field Descriptions 969
6-63. GPMC_IRQSTATUS Field Descriptions 970
6-64. GPMC_IRQENABLE Field Descriptions 971
6-65. GPMC_TIMEOUT_CONTROL Field Descriptions 972
6-66. GPMC_ERR_ADDRESS Field Descriptions 972
6-67. GPMC_ERR_TYPE Field Descriptions 973
6-68. GPMC_CONFIG Field Descriptions 974
6-69. GPMC_STATUS Field Descriptions 975
6-70. GPMC_CONFIG1_i Field Descriptions 976
6-71. GPMC_CONFIG2_i Field Descriptions 979
6-72. GPMC_CONFIG3_i Field Descriptions 980
6-73. GPMC_CONFIG4_i Field Descriptions 982
6-74. GPMC_CONFIG5_i Field Descriptions 984
6-75. GPMC_CONFIG6_i Field Descriptions 985
6-76. GPMC_CONFIG7_i Field Descriptions 986
6-77. GPMC_NAND_COMMAND_i Field Descriptions 987
6-78. GPMC_NAND_ADDRESS_i Field Descriptions 987
6-79. GPMC_NAND_DATA_i Field Descriptions 987
6-80. GPMC_PREFETCH_CONFIG1 Field Descriptions 988
6-81. GPMC_PREFETCH_CONFIG2 Field Descriptions 990
6-82. GPMC_PREFETCH_CONTROL Field Descriptions 990
6-83. GPMC_PREFETCH_STATUS Field Descriptions 991
6-84. GPMC_ECC_CONFIG Field Descriptions 992
6-85. GPMC_ECC_CONTROL Field Descriptions 993
6-86. GPMC_ECC_SIZE_CONFIG Field Descriptions 994
6-87. GPMC_ECCj_RESULT Field Descriptions 996
6-88. GPMC_BCH_RESULT0_i Field Descriptions 997
6-89. GPMC_BCH_RESULT1_i Field Descriptions 997
6-90. GPMC_BCH_RESULT2_i Field Descriptions 997
6-91. GPMC_BCH_RESULT3_i Field Descriptions 998
6-92. GPMC_BCH_SWDATA Field Descriptions 998
6-93. GPMC_BCH_RESULT4_i Field Descriptions 998
6-94. GPMC_BCH_RESULT5_i Field Descriptions 999
6-95. GPMC_BCH_RESULT6_i Field Descriptions 999
6-96. OCMC RAM Connectivity Attributes 1001
6-97. OCMC RAM Clock Signals 1001
6-98. Unsupported EMIF Features 1004
6-99. EMIF Connectivity Attributes 1005
6-100. EMIF Clock Signals 1005
6-101. EMIF Pin List 1005
6-102. EMIF4D REGISTERS 1007
6-103. EMIF_MOD_ID_REV Register Field Descriptions 1009
6-104. IODFT_CTRL_MISR_RSLT Register Field Descriptions 1010
6-105. IODFT_ADDR_MISR_RSLT Register Field Descriptions 1011
6-106. IODFT_DATA_MISR_RSLT_1 Register Field Descriptions 1012
6-107. IODFT_DATA_MISR_RSLT_2 Register Field Descriptions 1013

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6-108. IODFT_DATA_MISR_RSLT_3 Register Field Descriptions 1014
6-109. SDRAM_CONFIG_2 Register Field Descriptions 1015
6-110. PERF_CNT_1 Register Field Descriptions 1016
6-111. PERF_CNT_2 Register Field Descriptions 1017
6-112. PERF_CNT_CFG Register Field Descriptions 1018
6-113. PERF_CNT_SEL Register Field Descriptions 1019
6-114. PERF_CNT_TIM Register Field Descriptions 1020
6-115. READ_IDLE_CTRL Register Field Descriptions 1021
6-116. READ_IDLE_CTRL_SHDW Register Field Descriptions 1022
6-117. SDRAM_REF_CTRL Register Field Descriptions 1023
6-118. IRQ_EOI Register Field Descriptions 1024
6-119. IRQSTATUS_RAW_SYS Register Field Descriptions 1025
6-120. IRQSTATUS_RAW_LL Register Field Descriptions 1026
6-121. IRQSTATUS_SYS Register Field Descriptions 1027
6-122. IRQSTATUS_LL Register Field Descriptions 1028
6-123. IRQENABLE_SET_SYS Register Field Descriptions 1029
6-124. IRQENABLE_SET_LL Register Field Descriptions 1030
6-125. IRQENABLE_CLR_SYS Register Field Descriptions 1031
6-126. IRQENABLE_CLR_LL Register Field Descriptions 1032
6-127. SDRAM_REF_CTRL_SHDW Register Field Descriptions 1033
6-128. ZQ_CONFIG Register Field Descriptions 1034
6-129. TEMP_ALERT_CONFIG Register Field Descriptions 1035
6-130. OCP_ERR_LOG Register Field Descriptions 1036
6-131. DDR_PHY_CTRL_1 Register Field Descriptions 1037
6-132. DDR_PHY_CTRL_1_SHDW Register Field Descriptions 1038
6-133. DDR_PHY_CTRL_2 Register Field Descriptions 1039
6-134. SDRAM_TIM_1 Register Field Descriptions 1040
6-135. SDRAM_TIM_1_SHDW Register Field Descriptions 1041
6-136. SDRAM_TIM_2 Register Field Descriptions 1042
6-137. SDRAM_TIM_2_SHDW Register Field Descriptions 1043
6-138. STATUS Register Field Descriptions 1044
6-139. SDRAM_TIM_3 Register Field Descriptions 1045
6-140. SDRAM_TIM_3_SHDW Register Field Descriptions 1046
6-141. LPDDR2_NVM_TIM Register Field Descriptions 1047
6-142. LPDDR2_NVM_TIM_SHDW Register Field Descriptions 1048
6-143. PWR_MGMT_CTRL Register Field Descriptions 1049
6-144. PWR_MGMT_CTRL_SHDW Register Field Descriptions 1050
6-145. LPDDR2_MODE_REG_DATA Register Field Descriptions 1051
6-146. SDRAM_CONFIG Register Field Descriptions 1052
6-147. LPDDR2_MODE_REG_CFG Register Field Descriptions 1054
6-148. OCP_CONFIG Register Field Descriptions 1055
6-149. OCP_CFG_VAL_1 Register Field Descriptions 1056
6-150. OCP_CFG_VAL_2 Register Field Descriptions 1057
6-151. IODFT_TLGC Register Field Descriptions 1058
6-152. ELM Connectivity Attributes 1060
6-153. ELM Clock Signals 1060
6-154. Local Power Management Features 1061
6-155. Events 1061
6-156. ELM_LOCATION_STATUS_i Value Decoding Table 1063

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6-157. ELM Processing Initialization 1064
6-158. ELM Processing Completion for Continuous Mode 1064
6-159. ELM Processing Completion for Page Mode 1065
6-160. Use Case: Continuous Mode 1065
6-161. 16-bit NAND Sector Buffer Address Map 1067
6-162. Use Case: Page Mode 1067
6-163. ELM Instance Summary 1070
6-164. ELM Registers Mapping Summary 1070
6-165. ELM Revision Register (ELM_REVISION) Field Descriptions 1071
6-166. ELM System Configuration Register (ELM_SYSCONFIG) Field Descriptions 1071
6-167. ELM System Status Register (ELM_SYSSTATUS) Field Descriptions 1072
6-168. ELM Interrupt Status Register (ELM_IRQSTATUS) Field Descriptions 1073
6-169. ELM Interrupt Enable Register (ELM_IRQENABLE) Field Descriptions 1075
6-170. ELM Location Configuration Register (ELM_LOCATION_CONFIG) Field Descriptions 1076
6-171. ELM Page Definition Register (ELM_PAGE_CTRL) Field Descriptions 1077
6-172. ELM_SYNDROME_FRAGMENT_0_i Register Field Descriptions 1078
6-173. ELM_SYNDROME_FRAGMENT_1_i Register Field Descriptions 1078
6-174. ELM_SYNDROME_FRAGMENT_2_i Register Field Descriptions 1078
6-175. ELM_SYNDROME_FRAGMENT_3_i Register Field Descriptions 1079
6-176. ELM_SYNDROME_FRAGMENT_4_i Register Field Descriptions 1079
6-177. ELM_SYNDROME_FRAGMENT_5_i Register Field Descriptions 1079
6-178. ELM_SYNDROME_FRAGMENT_6_i Register Field Descriptions 1080
6-179. ELM_LOCATION_STATUS_i Register Field Descriptions 1080
6-180. ELM_ERROR_LOCATION_0-15_i Registers Field Descriptions 1081
7-1. PRM_CEFUSE REGISTERS 1085
7-2. PM_CEFUSE_PWRSTCTRL Register Field Descriptions 1086
7-3. PM_CEFUSE_PWRSTST Register Field Descriptions 1087
7-4. PRM_DEVICE REGISTERS 1088
7-5. PRM_RSTCTRL Register Field Descriptions 1089
7-6. PRM_SRAM_COUNT Register Field Descriptions 1090
7-7. PRM_LDO_SRAM_CORE_SETUP Register Field Descriptions 1091
7-8. PRM_LDO_SRAM_CORE_CTRL Register Field Descriptions 1092
7-9. PRM_LDO_SRAM_MPU_SETUP Register Field Descriptions 1093
7-10. PRM_LDO_SRAM_MPU_CTRL Register Field Descriptions 1094
7-11. PRM_RSTTIME Register Field Descriptions 1095
7-12. PRM_RSTST Register Field Descriptions 1096
7-13. PRM_MPU REGISTERS 1097
7-14. PM_MPU_PWRSTCTRL Register Field Descriptions 1098
7-15. PM_MPU_PWRSTST Register Field Descriptions 1099
7-16. RM_MPU_RSTST Register Field Descriptions 1100
7-17. PRM_PER REGISTERS 1101
7-18. RM_PER_RSTCTRL Register Field Descriptions 1102
7-19. PM_PER_PWRSTCTRL Register Field Descriptions 1103
7-20. RM_PER_RSTST Register Field Descriptions 1104
7-21. PM_PER_PWRSTST Register Field Descriptions 1105
7-22. PRM_RTC REGISTERS 1106
7-23. PM_RTC_PWRSTCTRL Register Field Descriptions 1107
7-24. PM_RTC_PWRSTST Register Field Descriptions 1108
7-25. PRM_WKUP REGISTERS 1109

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7-26. RM_WKUP_RSTCTRL Register Field Descriptions 1110
7-27. RM_WKUP_RSTST Register Field Descriptions 1111
7-28. PM_WKUP_PWRSTCTRL Register Field Descriptions 1112
7-29. PM_WKUP_PWRSTST Register Field Descriptions 1113
7-30. CM_CEFUSE REGISTERS 1114
7-31. CM_CEFUSE_CLKSTCTRL Register Field Descriptions 1115
7-32. CM_CEFUSE_CEFUSE_CLKCTRL Register Field Descriptions 1116
7-33. CM_DEVICE REGISTERS 1117
7-34. CM_CLKOUT_CTRL Register Field Descriptions 1118
7-35. CM_DPLL REGISTERS 1119
7-36. CLKSEL_TIMER7_CLK Register Field Descriptions 1120
7-37. CLKSEL_TIMER4_CLK Register Field Descriptions 1121
7-38. CM_MAC_CLKSEL Register Field Descriptions 1122
7-39. CLKSEL_TIMER5_CLK Register Field Descriptions 1123
7-40. CLKSEL_TIMER6_CLK Register Field Descriptions 1124
7-41. CM_CPTS_RFT_CLKSEL Register Field Descriptions 1125
7-42. CLKSEL_TIMER1MS_CLK Register Field Descriptions 1126
7-43. CLKSEL_TIMER2_CLK Register Field Descriptions 1127
7-44. CLKSEL_GFX_FCLK Register Field Descriptions 1128
7-45. CLKSEL_ICSS_OCP_CLK Register Field Descriptions 1129
7-46. CLKSEL_LCDC_PIXEL_CLK Register Field Descriptions 1130
7-47. CLKSEL_WDT1_CLK Register Field Descriptions 1131
7-48. CLKSEL_GPIO0_DBCLK Register Field Descriptions 1132
7-49. CLKSEL_TIMER3_CLK Register Field Descriptions 1133
7-50. CM_GFX REGISTERS 1133
7-51. CM_GFX_L3_CLKSTCTRL Register Field Descriptions 1134
7-52. CM_GFX_L4LS_GFX_CLKSTCTRL Register Field Descriptions 1135
7-53. CM_GFX_MMUCFG_CLKCTRL Register Field Descriptions 1136
7-54. CM_GFX_MMUDATA_CLKCTRL Register Field Descriptions 1137
7-55. CM_GFX_GFX_CLKCTRL Register Field Descriptions 1138
7-56. CM_GFX_BITBLT_CLKCTRL Register Field Descriptions 1139
7-57. CM_MPU REGISTERS 1139
7-58. CM_MPU_CLKSTCTRL Register Field Descriptions 1140
7-59. CM_MPU_MPU_CLKCTRL Register Field Descriptions 1141
7-60. CM_PER REGISTERS 1142
7-61. CM_PER_L4LS_CLKSTCTRL Register Field Descriptions 1146
7-62. CM_PER_L4FW_CLKCTRL Register Field Descriptions 1148
7-63. CM_PER_MCASP1_CLKCTRL Register Field Descriptions 1149
7-64. CM_PER_UART1_CLKCTRL Register Field Descriptions 1150
7-65. CM_PER_UART2_CLKCTRL Register Field Descriptions 1151
7-66. CM_PER_UART3_CLKCTRL Register Field Descriptions 1152
7-67. CM_PER_L3_CLKSTCTRL Register Field Descriptions 1153
7-68. CM_PER_UART4_CLKCTRL Register Field Descriptions 1154
7-69. CM_PER_TIMER7_CLKCTRL Register Field Descriptions 1155
7-70. CM_PER_TIMER2_CLKCTRL Register Field Descriptions 1156
7-71. CM_PER_TIMER3_CLKCTRL Register Field Descriptions 1157
7-72. CM_PER_TIMER4_CLKCTRL Register Field Descriptions 1158
7-73. CM_PER_MCASP2_CLKCTRL Register Field Descriptions 1159
7-74. CM_PER_RNG_CLKCTRL Register Field Descriptions 1160

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7-75. CM_PER_AES0_CLKCTRL Register Field Descriptions 1161
7-76. CM_PER_AES1_CLKCTRL Register Field Descriptions 1162
7-77. CM_PER_DES_CLKCTRL Register Field Descriptions 1163
7-78. Reserved Register Field Descriptions 1164
7-79. CM_PER_SHA0_CLKCTRL Register Field Descriptions 1165
7-80. CM_PER_PKA_CLKCTRL Register Field Descriptions 1166
7-81. CM_PER_GPIO6_CLKCTRL Register Field Descriptions 1167
7-82. CM_PER_GPIO1_CLKCTRL Register Field Descriptions 1168
7-83. CM_PER_GPIO2_CLKCTRL Register Field Descriptions 1169
7-84. CM_PER_GPIO3_CLKCTRL Register Field Descriptions 1170
7-85. CM_PER_GPIO4_CLKCTRL Register Field Descriptions 1171
7-86. CM_PER_TPCC_CLKCTRL Register Field Descriptions 1172
7-87. CM_PER_DCAN0_CLKCTRL Register Field Descriptions 1173
7-88. CM_PER_DCAN1_CLKCTRL Register Field Descriptions 1174
7-89. CM_PER_CPGMAC0_CLKCTRL Register Field Descriptions 1175
7-90. CM_PER_SPARE_CLKCTRL Register Field Descriptions 1176
7-91. CM_PER_EPWMSS1_CLKCTRL Register Field Descriptions 1177
7-92. CM_PER_EMIF_FW_CLKCTRL Register Field Descriptions 1178
7-93. CM_PER_EPWMSS0_CLKCTRL Register Field Descriptions 1179
7-94. CM_PER_EPWMSS2_CLKCTRL Register Field Descriptions 1180
7-95. CM_PER_L3_INSTR_CLKCTRL Register Field Descriptions 1181
7-96. CM_PER_L3_CLKCTRL Register Field Descriptions 1182
7-97. CM_PER_IEEE5000_CLKCTRL Register Field Descriptions 1183
7-98. CM_PER_ICSS_CLKCTRL Register Field Descriptions 1184
7-99. CM_PER_TIMER5_CLKCTRL Register Field Descriptions 1185
7-100. CM_PER_LCDC_CLKCTRL Register Field Descriptions 1186
7-101. CM_PER_TIMER6_CLKCTRL Register Field Descriptions 1187
7-102. CM_PER_MMC1_CLKCTRL Register Field Descriptions 1188
7-103. CM_PER_MMC2_CLKCTRL Register Field Descriptions 1189
7-104. CM_PER_TPTC1_CLKCTRL Register Field Descriptions 1190
7-105. CM_PER_TPTC2_CLKCTRL Register Field Descriptions 1191
7-106. CM_PER_GPIO5_CLKCTRL Register Field Descriptions 1192
7-107. CM_PER_SPINLOCK_CLKCTRL Register Field Descriptions 1193
7-108. CM_PER_MAILBOX0_CLKCTRL Register Field Descriptions 1194
7-109. CM_PER_USB0_CLKCTRL Register Field Descriptions 1195
7-110. CM_PER_L4HS_CLKSTCTRL Register Field Descriptions 1196
7-111. CM_PER_L4HS_CLKCTRL Register Field Descriptions 1197
7-112. CM_PER_MSTR_EXPS_CLKCTRL Register Field Descriptions 1198
7-113. CM_PER_SLV_EXPS_CLKCTRL Register Field Descriptions 1199
7-114. CM_PER_OCPWP_L3_CLKSTCTRL Register Field Descriptions 1200
7-115. CM_PER_OCPWP_CLKCTRL Register Field Descriptions 1201
7-116. CM_PER_MAILBOX1_CLKCTRL Register Field Descriptions 1202
7-117. CM_PER_SPARE0_CLKCTRL Register Field Descriptions 1203
7-118. CM_PER_SPARE1_CLKCTRL Register Field Descriptions 1204
7-119. CM_PER_MLB_CLKCTRL Register Field Descriptions 1205
7-120. CM_PER_ICSS_CLKSTCTRL Register Field Descriptions 1206
7-121. CM_PER_CPSW_CLKSTCTRL Register Field Descriptions 1207
7-122. CM_PER_LCDC_CLKSTCTRL Register Field Descriptions 1208
7-123. CM_PER_CLKDIV32K_CLKCTRL Register Field Descriptions 1209

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7-124. CM_PER_CLK_24MHZ_CLKSTCTRL Register Field Descriptions 1210
7-125. CM_PER_TPTC0_CLKCTRL Register Field Descriptions 1211
7-126. CM_PER_L3S_CLKSTCTRL Register Field Descriptions 1212
7-127. CM_PER_EMIF_CLKCTRL Register Field Descriptions 1213
7-128. CM_PER_OCMCRAM_CLKCTRL Register Field Descriptions 1214
7-129. CM_PER_GPMC_CLKCTRL Register Field Descriptions 1215
7-130. CM_PER_MCASP0_CLKCTRL Register Field Descriptions 1216
7-131. CM_PER_UART5_CLKCTRL Register Field Descriptions 1217
7-132. CM_PER_MMC0_CLKCTRL Register Field Descriptions 1218
7-133. CM_PER_ELM_CLKCTRL Register Field Descriptions 1219
7-134. CM_PER_I2C2_CLKCTRL Register Field Descriptions 1220
7-135. CM_PER_I2C1_CLKCTRL Register Field Descriptions 1221
7-136. CM_PER_SPI0_CLKCTRL Register Field Descriptions 1222
7-137. CM_PER_L4FW_CLKSTCTRL Register Field Descriptions 1223
7-138. CM_PER_SPI1_CLKCTRL Register Field Descriptions 1224
7-139. CM_PER_SPI2_CLKCTRL Register Field Descriptions 1225
7-140. CM_PER_SPI3_CLKCTRL Register Field Descriptions 1226
7-141. CM_PER_L4LS_CLKCTRL Register Field Descriptions 1227
7-142. CM_RTC REGISTERS 1227
7-143. CM_RTC_RTC_CLKCTRL Register Field Descriptions 1228
7-144. CM_RTC_CLKSTCTRL Register Field Descriptions 1229
7-145. CM_WKUP REGISTERS 1230
7-146. CM_WKUP_CLKSTCTRL Register Field Descriptions 1233
7-147. CM_SSC_MODFREQDIV_DPLL_CORE Register Field Descriptions 1234
7-148. CM_CLKSEL_DPLL_CORE Register Field Descriptions 1235
7-149. CM_AUTOIDLE_DPLL_PER Register Field Descriptions 1236
7-150. CM_IDLEST_DPLL_PER Register Field Descriptions 1237
7-151. CM_SSC_DELTAMSTEP_DPLL_PER Register Field Descriptions 1238
7-152. CM_WKUP_L4WKUP_CLKCTRL Register Field Descriptions 1239
7-153. CM_SSC_MODFREQDIV_DPLL_PER Register Field Descriptions 1240
7-154. CM_CLKDCOLDO_DPLL_PER Register Field Descriptions 1241
7-155. CM_DIV_M4_DPLL_CORE Register Field Descriptions 1242
7-156. CM_DIV_M5_DPLL_CORE Register Field Descriptions 1243
7-157. CM_CLKMODE_DPLL_MPU Register Field Descriptions 1244
7-158. CM_CLKMODE_DPLL_PER Register Field Descriptions 1246
7-159. CM_CLKMODE_DPLL_CORE Register Field Descriptions 1247
7-160. CM_CLKMODE_DPLL_DDR Register Field Descriptions 1249
7-161. CM_CLKMODE_DPLL_DISP Register Field Descriptions 1251
7-162. CM_CLKSEL_DPLL_PERIPH Register Field Descriptions 1253
7-163. CM_WKUP_TIMER0_CLKCTRL Register Field Descriptions 1254
7-164. CM_DIV_M2_DPLL_DDR Register Field Descriptions 1255
7-165. CM_DIV_M2_DPLL_DISP Register Field Descriptions 1256
7-166. CM_DIV_M2_DPLL_MPU Register Field Descriptions 1257
7-167. CM_DIV_M2_DPLL_PER Register Field Descriptions 1258
7-168. CM_WKUP_WKUP_M3_CLKCTRL Register Field Descriptions 1259
7-169. CM_WKUP_UART0_CLKCTRL Register Field Descriptions 1260
7-170. CM_WKUP_I2C0_CLKCTRL Register Field Descriptions 1261
7-171. CM_WKUP_ADC_TSC_CLKCTRL Register Field Descriptions 1262
7-172. CM_WKUP_SMARTREFLEX0_CLKCTRL Register Field Descriptions 1263

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7-173. CM_WKUP_TIMER1_CLKCTRL Register Field Descriptions 1264
7-174. CM_WKUP_DEBUGSS_CLKCTRL Register Field Descriptions 1265
7-175. CM_WKUP_SMARTREFLEX1_CLKCTRL Register Field Descriptions 1266
7-176. CM_L4_WKUP_AON_CLKSTCTRL Register Field Descriptions 1267
7-177. CM_WKUP_WDT0_CLKCTRL Register Field Descriptions 1268
7-178. CM_WKUP_WDT1_CLKCTRL Register Field Descriptions 1269
7-179. CM_DIV_M6_DPLL_CORE Register Field Descriptions 1270
7-180. CM_L3_AON_CLKSTCTRL Register Field Descriptions 1271
7-181. CM_AUTOIDLE_DPLL_MPU Register Field Descriptions 1272
7-182. CM_IDLEST_DPLL_MPU Register Field Descriptions 1273
7-183. CM_SSC_DELTAMSTEP_DPLL_MPU Register Field Descriptions 1274
7-184. CM_WKUP_CONTROL_CLKCTRL Register Field Descriptions 1275
7-185. CM_SSC_MODFREQDIV_DPLL_MPU Register Field Descriptions 1276
7-186. CM_CLKSEL_DPLL_MPU Register Field Descriptions 1277
7-187. CM_AUTOIDLE_DPLL_DDR Register Field Descriptions 1278
7-188. CM_IDLEST_DPLL_DDR Register Field Descriptions 1279
7-189. CM_SSC_DELTAMSTEP_DPLL_DDR Register Field Descriptions 1280
7-190. CM_SSC_MODFREQDIV_DPLL_DDR Register Field Descriptions 1281
7-191. CM_CLKSEL_DPLL_DDR Register Field Descriptions 1282
7-192. CM_AUTOIDLE_DPLL_DISP Register Field Descriptions 1283
7-193. CM_IDLEST_DPLL_DISP Register Field Descriptions 1284
7-194. CM_SSC_DELTAMSTEP_DPLL_DISP Register Field Descriptions 1285
7-195. CM_WKUP_GPIO0_CLKCTRL Register Field Descriptions 1286
7-196. CM_SSC_MODFREQDIV_DPLL_DISP Register Field Descriptions 1287
7-197. CM_CLKSEL_DPLL_DISP Register Field Descriptions 1288
7-198. CM_AUTOIDLE_DPLL_CORE Register Field Descriptions 1289
7-199. CM_IDLEST_DPLL_CORE Register Field Descriptions 1290
7-200. CM_SSC_DELTAMSTEP_DPLL_CORE Register Field Descriptions 1291
9-1. Instance Summary 1297
9-2. Initiator Agent Common Register Mapping Summary 1298
9-3. Initiator Agent Common Register Mapping Summary 1298
9-4. Initiator Agent Common Register Mapping Summary 1298
9-5. Initiator Agent Common Register Mapping Summary 1299
9-6. Agent Control Register - IA Block (L3_IA_AGENT_CONTROL) Field Descriptions 1300
9-7. Agent Status Register (L3_IA_AGENT_STATUS) Field Descriptions 1302
9-8. Error Log Register - IA Block (L3_IA_ERROR_LOG) Field Descriptions 1304
9-9. Error Log Address Register - IA Block (L3_IA_ERROR_LOG_ADDR) Field Descriptions 1305
9-10. Target Agent Common Register Mapping Summary 1306
9-11. Target Agent Common Register Mapping Summary 1306
9-12. Target Agent Common Register Mapping Summary 1306
9-13. Target Agent Common Register Mapping Summary 1306
9-14. Agent Control Register - TA Block (L3_TA_AGENT_CONTROL) Field Descriptions 1308
9-15. Agent Status Register (L3_TA_AGENT_STATUS) Field Descriptions 1309
9-16. Error Log Register - TA Block (L3_TA_ERROR_LOG) Field Descriptions 1310
9-17. Error Log Address Register - TA Block (L3_TA_ERROR_LOG_ADDR) Field Descriptions 1311
9-18. RT Register Mapping Summary 1312
9-19. L3_RT_NETWORK Field Descriptions 1313
9-20. L3_RT_INITID_READBACK Field Descriptions 1313
9-21. L3_RT_NETWORK_CONTROL Field Descriptions 1314

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9-22. Protection Mechanism Common Register Mapping Summary 1315
9-23. Protection Mechanism Common Register Mapping Summary 1315
9-24. Protection Mechanism Common Register Mapping Summary 1316
9-25. L3_PM_ERROR_LOG Field Descriptions 1317
9-26. L3_PM_CONTROL Field Descriptions 1318
9-27. L3_PM_ERROR_CLEAR_SINGLE Field Descriptions 1319
9-28. L3_PM_ERROR_CLEAR_MULTI Field Descriptions 1319
9-29. L3_PM_REQ_INFO_PERMISSION_i Field Descriptions 1320
9-30. Reset Value for REQ_INFO_PERMISSION 1320
9-31. L3_PM_READ_PERMISSION_i Field Descriptions 1321
9-32. L3_PM_WRITE_PERMISSION_i Field Descriptions 1322
9-33. Bit Availability and Initialization Values for L3_PM_READ_PERMISSION_i and L3_PM_WRITE_PERMISSION_i 1323
9-34. L3_PM_ADDR_MATCH_k Field Descriptions 1324
9-35. Reset Value for L3_PM_ADDR_MATCH_k 1324
9-36. SI Register Mapping Summary 1326
9-37. L3_SI_CONTROL Field Descriptions 1327
9-38. L3_SI_FLAG_STATUS_0 Field Descriptions 1328
9-39. L3_SI_FLAG_STATUS_1 Field Descriptions 1328
9-40. L4- Core Instance Summary 1329
9-41. L4-Per Instance Summary 1330
9-42. L4-Emu Instance Summary 1330
9-43. L4-WKUP Instance Summary 1330
9-44. L4 IA Register Mapping Summary (1) 1332
9-45. L4 IA Register Mapping Summary (2) 1332
9-46. L4_IA_AGENT_CONTROL_L Field Descriptions 1333
9-47. L4_IA_AGENT_STATUS_L Field Descriptions 1334
9-48. L4_IA_ERROR_LOG_L Field Descriptions 1334
9-49. CORE_TA Common Register Mapping Summary 1335
9-50. CORE_TA Common Register Mapping Summary 1335
9-51. CORE_TA Common Register Mapping Summary 1335
9-52. CORE_TA Common Register Mapping Summary 1335
9-53. CORE_TA Common Register Mapping Summary 1336
9-54. CORE_TA Common Register Mapping Summary 1336
9-55. CORE_TA Common Register Mapping Summary 1336
9-56. CORE_TA Common Register Mapping Summary 1336
9-57. CORE_TA Common Register Mapping Summary 1336
9-58. CORE_TA Common Register Mapping Summary 1337
9-59. CORE_TA Common Register Mapping Summary 1337
9-60. CORE_TA Common Register Mapping Summary 1337
9-61. CORE_TA Common Register Mapping Summary 1337
9-62. CORE_TA Common Register Mapping Summary 1337
9-63. CORE_TA Common Register Mapping Summary 1338
9-64. CORE_TA Common Register Mapping Summary 1338
9-65. CORE_TA Common Register Mapping Summary 1338
9-66. CORE_TA Common Register Mapping Summary 1338
9-67. PER_TA Common Register Mapping Summary 1338
9-68. PER_TA Common Register Mapping Summary 1338
9-69. PER_TA Common Register Mapping Summary 1339

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9-70. PER_TA Common Register Mapping Summary 1339
9-71. PER_TA Common Register Mapping Summary 1339
9-72. PER_TA Common Register Mapping Summary 1339
9-73. PER_TA Common Register Mapping Summary 1339
9-74. EMU_TA Common Register Mapping Summary 1340
9-75. EMU_TA Common Register Mapping Summary 1340
9-76. WKUP_TA Common Register Mapping Summary 1340
9-77. WKUP_TA Common Register Mapping Summary 1341
9-78. WKUP_TA Common Register Mapping Summary 1341
9-79. L4_TA_AGENT_CONTROL_L Field Descriptions 1342
9-80. L4_TA_AGENT_CONTROL_H Field Descriptions 1343
9-81. L4_TA_AGENT_STATUS_L Field Descriptions 1343
9-82. L4 LA Register Mapping Summary 1344
9-83. L4_LA_NETWORK_H Field Descriptions 1345
9-84. L4_LA_INITIATOR_INFO_L Field Descriptions 1345
9-85. Reset value for L4_LA_INITIATOR_INFO_L 1345
9-86. L4_LA_INITIATOR_INFO_H Field Descriptions 1346
9-87. Reset value for L4_LA_INITIATOR_INFO_H 1346
9-88. L4_LA_NETWORK_CONTROL_L Field Descriptions 1347
9-89. L4_LA_NETWORK_CONTROL_H Field Descriptions 1348
9-90. L4 AP Register Mapping Summary 1349
9-91. L4 AP Register Mapping Summary 1349
9-92. Reset Values for CORE_AP L4_AP_REGION_l_L and L4_AP_REGION_l_H 1350
9-93. Reset Values for PER_AP L4_AP_REGION_l_L and L4_AP_REGION_l_H 1352
9-94. Reset Values for EMU_AP L4_AP_REGION_l_L and L4_AP_REGION_l_H 1353
9-95. Reset Values for WKPUP_AP L4_AP_REGION_l_L and L4_AP_REGION_l_H 1354
9-96. L4_AP_SEGMENT_i_L Field Descriptions 1355
9-97. Reset Values for L4_AP_SEGMENT_i_L 1355
9-98. L4_AP_SEGMENT_i_H Field Descriptions 1356
9-99. Reset Values for L4_AP_SEGMENT_i_H 1356
9-100. L4_AP_PROT_GROUP_MEMBERS_k_L Field Descriptions 1356
9-101. L4_AP_PROT_GROUP_ROLES_k_L Field Descriptions 1357
9-102. L4_AP_REGION_l_L Field Descriptions 1357
9-103. L4_AP_REGION_l_H Field Descriptions 1358
10-1. EDMACC Registers 1361
10-2. Peripheral ID Register (PID) Field Descriptions 1364
10-3. EDMA3CC Configuration Register (CCCFG) Field Descriptions 1365
10-4. DMA Channel Map *n* Registers (DCHMAP*n*) Field Descriptions 1367
10-5. QDMA Channel Map *n* Registers (QCHMAP*n*) Field Descriptions 1368
10-6. DMA Channel Queue *n* Number Registers (DMAQNUM*n*) Field Descriptions 1369
10-7. Bits in DMAQNUM*n* 1369
10-8. QDMA Channel Queue Number Register (QDMAQNUM) Field Descriptions 1370
10-9. Queue Priority Register (QUEPRI) Field Descriptions 1371
10-10. Event Missed Register (EMR) Field Descriptions 1372
10-11. Event Missed Register High (EMRH) Field Descriptions 1372
10-12. Event Missed Clear Register (EMCR) Field Descriptions 1373
10-13. Event Missed Clear Register High (EMCRH) Field Descriptions 1373
10-14. QDMA Event Missed Register (QEMR) Field Descriptions 1374
10-15. QDMA Event Missed Clear Register (QEMCR) Field Descriptions 1375

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10-16. EDMA3CC Error Register (CCERR) Field Descriptions 1376
10-17. EDMA3CC Error Clear Register (CCERRCLR) Field Descriptions 1376
10-18. Error Evaluation Register (EEVAL) Field Descriptions 1378
10-19. DMA Region Access Enable Registers for Region M (DRAE*m*/DRAEH*m*) Field Descriptions 1379
10-20. QDMA Region Access Enable for Region M (QRAE*m*) Field Descriptions 1380
10-21. Event Queue Entry Registers (Q*x*E*y*) Field Descriptions 1381
10-22. Queue Status Register *n* (QSTAT*n*) Field Descriptions 1382
10-23. Queue Watermark Threshold A Register (QWMTHRA) Field Descriptions 1383
10-24. EDMA3CC Status Register (CCSTAT) Field Descriptions 1384
10-25. Memory Protection Fault Address Register (MPFAR) Field Descriptions 1386
10-26. Memory Protection Fault Status Register (MPFSR) Field Descriptions 1387
10-27. Memory Protection Fault Command Register (MPFCR) Field Descriptions 1388
10-28. Memory Protection Page Attribute Register (MPPA*n*) Field Descriptions 1389
10-29. Event Register (ER) Field Descriptions 1391
10-30. Event Register High (ERH) Field Descriptions 1391
10-31. Event Clear Register (ECR) Field Descriptions 1392
10-32. Event Clear Register High (ECRH) Field Descriptions 1392
10-33. Event Set Register (ESR) Field Descriptions 1393
10-34. Event Set Register High (ESRH) Field Descriptions 1394
10-35. Chained Event Register (CER) Field Descriptions 1395
10-36. Chained Event Register High (CERH) Field Descriptions 1396
10-37. Event Enable Register (EER) Field Descriptions 1397
10-38. Event Enable Register High (EERH) Field Descriptions 1397
10-39. Event Enable Clear Register (EECR) Field Descriptions 1398
10-40. Event Enable Clear Register High (EECRH) Field Descriptions 1398
10-41. Event Enable Set Register (EESR) Field Descriptions 1399
10-42. Event Enable Set Register High (EESRH) Field Descriptions 1399
10-43. Secondary Event Register (SER) Field Descriptions 1400
10-44. Secondary Event Register High (SERH) Field Descriptions 1400
10-45. Secondary Event Clear Register (SECR) Field Descriptions 1401
10-46. Secondary Event Clear Register High (SECRH) Field Descriptions 1401
10-47. Interrupt Enable Register (IER) Field Descriptions 1402
10-48. Interrupt Enable Register High (IERH) Field Descriptions 1402
10-49. Interrupt Enable Clear Register (IECR) Field Descriptions 1403
10-50. Interrupt Enable Clear Register High (IECRH) Field Descriptions 1403
10-51. Interrupt Enable Set Register (IESR) Field Descriptions 1404
10-52. Interrupt Enable Set Register High (IESRH) Field Descriptions 1404
10-53. Interrupt Pending Register (IPR) Field Descriptions 1405
10-54. Interrupt Pending Register High (IPRH) Field Descriptions 1405
10-55. Interrupt Clear Register (ICR) Field Descriptions 1406
10-56. Interrupt Clear Register High (ICRH) Field Descriptions 1406
10-57. Interrupt Evaluate Register (IEVAL) Field Descriptions 1407
10-58. QDMA Event Register (QER) Field Descriptions 1408
10-59. QDMA Event Enable Register (QEER) Field Descriptions 1409
10-60. QDMA Event Enable Clear Register (QEECR) Field Descriptions 1410
10-61. QDMA Event Enable Set Register (QEESR) Field Descriptions 1411
10-62. QDMA Secondary Event Register (QSER) Field Descriptions 1412
10-63. QDMA Secondary Event Clear Register (QSECR) Field Descriptions 1413
10-64. EDMA3 Transfer Controller Registers 1414

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10-65. Peripheral ID Register (PID) Field Descriptions 1415
10-66. EDMA3TC Configuration Register (TCCFG) Field Descriptions 1416
10-67. EDMA3TC Channel Status Register (TCSTAT) Field Descriptions 1417
10-68. Error Register (ERRSTAT) Field Descriptions 1419
10-69. Error Enable Register (ERREN) Field Descriptions 1420
10-70. Error Clear Register (ERRCLR) Field Descriptions 1421
10-71. Error Details Register (ERRDET) Field Descriptions 1422
10-72. Error Interrupt Command Register (ERRCMD) Field Descriptions 1423
10-73. Read Rate Register (RDRATE) Field Descriptions 1424
10-74. Source Active Options Register (SAOPT) Field Descriptions 1425
10-75. Source Active Source Address Register (SASRC) Field Descriptions 1427
10-76. Source Active Count Register (SACNT) Field Descriptions 1427
10-77. Source Active Destination Address Register (SADST) Field Descriptions 1428
10-78. Source Active Source B-Dimension Index Register (SABIDX) Field Descriptions 1428
10-79. Source Active Memory Protection Proxy Register (SAMPPRXY) Field Descriptions 1429
10-80. Source Active Count Reload Register (SACNTRLD) Field Descriptions 1430
10-81. Source Active Source Address B-Reference Register (SASRCBREF) Field Descriptions 1430
10-82. Source Active Destination Address B-Reference Register (SADSTBREF) Field Descriptions 1431
10-83. Destination FIFO Options Register (DFOPT*n*) Field Descriptions 1432
10-84. Destination FIFO Source Address Register (DFSRC*n*) Field Descriptions 1434
10-85. Destination FIFO Count Register (DFCNT*n*) Field Descriptions 1434
10-86. Destination FIFO Destination Address Register (DFDST*n*) Field Descriptions 1435
10-87. Destination FIFO B-Index Register (DFBIDX*n*) Field Descriptions 1435
10-88. Destination FIFO Memory Protection Proxy Register (DFMPPRXY*n*) Field Descriptions 1436
10-89. Destination FIFO Count Reload Register (DFCNTRLD*n*) Field Descriptions 1437
10-90. Destination FIFO Source Address B-Reference Register (DFSRCBREF*n*) Field Descriptions 1437
10-91. Destination FIFO Destination Address B-Reference Register (DFDSTBREF*n*) Field Descriptions 1438
11-1. TSC_ADC Connectivity Attributes 1441
11-2. TSC_ADC Clock Signals 1442
11-3. TSC_ADC Pin List 1442
11-4. AFE I/O List 1444
11-5. TSC_ADC_SS REGISTERS 1447
11-6. REVISION Register Field Descriptions 1450
11-7. STEPCONFIG4 Register Field Descriptions 1451
11-8. STEPDELAY4 Register Field Descriptions 1452
11-9. STEPCONFIG5 Register Field Descriptions 1453
11-10. STEPDELAY5 Register Field Descriptions 1454
11-11. STEPCONFIG6 Register Field Descriptions 1455
11-12. IRQSTATUS_RAW Register Field Descriptions 1456
11-13. STEPDELAY6 Register Field Descriptions 1458
11-14. STEPCONFIG7 Register Field Descriptions 1459
11-15. STEPDELAY7 Register Field Descriptions 1460
11-16. STEPCONFIG8 Register Field Descriptions 1461
11-17. STEPDELAY8 Register Field Descriptions 1462
11-18. STEPCONFIG9 Register Field Descriptions 1463
11-19. STEPDELAY9 Register Field Descriptions 1464
11-20. STEPCONFIG10 Register Field Descriptions 1465
11-21. STEPDELAY10 Register Field Descriptions 1466
11-22. STEPCONFIG11 Register Field Descriptions 1467

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11-23. IRQSTATUS Register Field Descriptions 1468
11-24. STEPDELAY11 Register Field Descriptions 1470
11-25. STEPCONFIG12 Register Field Descriptions 1471
11-26. STEPDELAY12 Register Field Descriptions 1472
11-27. STEPCONFIG13 Register Field Descriptions 1473
11-28. STEPDELAY13 Register Field Descriptions 1474
11-29. STEPCONFIG14 Register Field Descriptions 1475
11-30. STEPDELAY14 Register Field Descriptions 1476
11-31. STEPCONFIG15 Register Field Descriptions 1477
11-32. STEPDELAY15 Register Field Descriptions 1478
11-33. STEPCONFIG16 Register Field Descriptions 1479
11-34. IRQENABLE_SET Register Field Descriptions 1480
11-35. STEPDELAY16 Register Field Descriptions 1482
11-36. FIFO0COUNT Register Field Descriptions 1483
11-37. FIFO0THRESHOLD Register Field Descriptions 1484
11-38. DMA0REQ Register Field Descriptions 1485
11-39. FIFO1COUNT Register Field Descriptions 1486
11-40. FIFO1THRESHOLD Register Field Descriptions 1487
11-41. DMA1REQ Register Field Descriptions 1488
11-42. FIFO0DATA Register Field Descriptions 1489
11-43. FIFO1DATA Register Field Descriptions 1490
11-44. IRQENABLE_CLR Register Field Descriptions 1491
11-45. IRQWAKEUP Register Field Descriptions 1493
11-46. DMAENABLE_SET Register Field Descriptions 1494
11-47. DMAENABLE_CLR Register Field Descriptions 1495
11-48. SYSCONFIG Register Field Descriptions 1496
11-49. CTRL Register Field Descriptions 1497
11-50. ADCSTAT Register Field Descriptions 1498
11-51. ADCRANGE Register Field Descriptions 1499
11-52. ADC_CLKDIV Register Field Descriptions 1500
11-53. ADC_MISC Register Field Descriptions 1501
11-54. STEPENABLE Register Field Descriptions 1502
11-55. IDLECONFIG Register Field Descriptions 1503
11-56. TS_CHARGE_STEPCONFIG Register Field Descriptions 1504
11-57. TS_CHARGE_DELAY Register Field Descriptions 1505
11-58. STEPCONFIG1 Register Field Descriptions 1506
11-59. IRQ_EOI Register Field Descriptions 1507
11-60. STEPDELAY1 Register Field Descriptions 1508
11-61. STEPCONFIG2 Register Field Descriptions 1509
11-62. STEPDELAY2 Register Field Descriptions 1510
11-63. STEPCONFIG3 Register Field Descriptions 1511
11-64. STEPDELAY3 Register Field Descriptions 1512
12-1. LCD Controller Connectivity Attributes 1516
12-2. LCD Controller Clock Signals 1517
12-3. LCD Controller Pin List 1517
12-4. LCD External I/O Signals 1520
12-5. Register Configuration for DMA Engine Programming 1521
12-6. LIDD I/O Name Map 1523
12-7. Operation Modes Supported by Raster Controller 1527

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12-8. Bits-Per-Pixel Encoding for Palette Entry 0 Buffer 1529
12-9. Frame Buffer Size According to BPP 1530
12-10. Color/Grayscale Intensities and Modulation Rates 1534
12-11. Number of Colors/Shades of Gray Available on Screen 1534
12-12. LCD REGISTERS 1539
12-13. pid Register Field Descriptions 1540
12-14. IRQENABLE_CLEAR Register Field Descriptions 1541
12-15. IRQEOI_VECTOR Register Field Descriptions 1543
12-16. CLKC_ENABLE Register Field Descriptions 1544
12-17. CLKC_RESET Register Field Descriptions 1545
12-18. LIDD_CTRL Register Field Descriptions 1546
12-19. LIDD_CS0_CONF Register Field Descriptions 1548
12-20. LIDD_CS0_ADDR Register Field Descriptions 1549
12-21. reg29 Register Field Descriptions 1550
12-22. reg30 Register Field Descriptions 1551
12-23. LIDD_CS0_DATA Register Field Descriptions 1552
12-24. reg31 Register Field Descriptions 1553
12-25. LIDD_CS1_CONF Register Field Descriptions 1554
12-26. LIDD_CS1_ADDR Register Field Descriptions 1555
12-27. LIDD_CS1_DATA Register Field Descriptions 1556
12-28. CTRL Register Field Descriptions 1557
12-29. RASTER_CTRL Register Field Descriptions 1558
12-30. RASTER_TIMING_0 Register Field Descriptions 1560
12-31. RASTER_TIMING_1 Register Field Descriptions 1561
12-32. RASTER_TIMING_2 Register Field Descriptions 1562
12-33. RASTER_SUBPANEL Register Field Descriptions 1564
12-34. RASTER_SUBPANEL2 Register Field Descriptions 1565
12-35. LCDDMA_CTRL Register Field Descriptions 1566
12-36. LCDDMA_FB0_BASE Register Field Descriptions 1567
12-37. LCDDMA_FB0_CEILING Register Field Descriptions 1568
12-38. LCDDMA_FB1_BASE Register Field Descriptions 1569
12-39. LCDDMA_FB1_CEILING Register Field Descriptions 1570
12-40. SYSCONFIG Register Field Descriptions 1571
12-41. IRQSTATUS_RAW Register Field Descriptions 1572
12-42. IRQSTATUS Register Field Descriptions 1573
12-43. IRQENABLE_SET Register Field Descriptions 1574
13-1. Unsupported CPGMAC Features 1579
13-2. Ethernet Switch Connectivity Attributes 1580
13-3. Ethernet Switch Clock Signals 1581
13-4. Ethernet Switch Pin List 1581
13-5. Learned Address Control Bits 1586
13-6. Free (Unused) Address Table Entry Bit Values 1586
13-7. Multicast Address Table Entry Bit Values 1586
13-8. VLAN/Multicast Address Table Entry Bit Values 1587
13-9. Unicast Address Table Entry Bit Values 1588
13-10. OUI Unicast Address Table Entry Bit Values 1589
13-11. Unicast Address Table Entry Bit Values 1589
13-12. VLAN Table Entry 1590
13-13. GMII Interface Signal Descriptions in GIG (1000Mbps) Mode 1597

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13-14. GMII Interface Signal Descriptions in MII (100/10Mbps) Mode 1597
13-15. 1599
13-16. 1601
13-17. Operations of Emulation Control Input and Register Bits 1605
13-18. Values of messageType field 1610
13-19. MDIO Read Frame Format 1615
13-20. MDIO Write Frame Format 1615
13-21. CPSW_ALE REGISTERS 1629
13-22. IDVER Register Field Descriptions 1630
13-23. PRESCALE Register Field Descriptions 1631
13-24. UNKNOWN_VLAN Register Field Descriptions 1632
13-25. TBLCTL Register Field Descriptions 1633
13-26. TBLW2 Register Field Descriptions 1634
13-27. TBLW1 Register Field Descriptions 1635
13-28. TBLW0 Register Field Descriptions 1636
13-29. PORTCTL0 Register Field Descriptions 1637
13-30. PORTCTL1 Register Field Descriptions 1638
13-31. PORTCTL2 Register Field Descriptions 1639
13-32. PORTCTL3 Register Field Descriptions 1640
13-33. CONTROL Register Field Descriptions 1641
13-34. PORTCTL4 Register Field Descriptions 1643
13-35. PORTCTL5 Register Field Descriptions 1644
13-36. CPSW_CPDMA REGISTERS 1645
13-37. TX_IDVER Register Field Descriptions 1648
13-38. TX_INTSTAT_RAW Register Field Descriptions 1649
13-39. TX_INTSTAT_MASKED Register Field Descriptions 1650
13-40. TX_INTMASK_SET Register Field Descriptions 1651
13-41. TX_INTMASK_CLEAR Register Field Descriptions 1652
13-42. CPDMA_IN_VECTOR Register Field Descriptions 1653
13-43. CPDMA_EOI_VECTOR Register Field Descriptions 1654
13-44. RX_IDVER Register Field Descriptions 1655
13-45. RX_INTSTAT_RAW Register Field Descriptions 1656
13-46. RX_INTSTAT_MASKED Register Field Descriptions 1657
13-47. RX_INTMASK_SET Register Field Descriptions 1658
13-48. RX_INTMASK_CLEAR Register Field Descriptions 1659
13-49. DMA_INTSTAT_RAW Register Field Descriptions 1660
13-50. DMA_INTSTAT_MASKED Register Field Descriptions 1661
13-51. DMA_INTMASK_SET Register Field Descriptions 1662
13-52. DMA_INTMASK_CLEAR Register Field Descriptions 1663
13-53. RX0_PENDTHRESH Register Field Descriptions 1664
13-54. RX1_PENDTHRESH Register Field Descriptions 1665
13-55. RX_CONTROL Register Field Descriptions 1666
13-56. RX2_PENDTHRESH Register Field Descriptions 1667
13-57. RX3_PENDTHRESH Register Field Descriptions 1668
13-58. RX4_PENDTHRESH Register Field Descriptions 1669
13-59. RX5_PENDTHRESH Register Field Descriptions 1670
13-60. RX6_PENDTHRESH Register Field Descriptions 1671
13-61. RX7_PENDTHRESH Register Field Descriptions 1672
13-62. RX0_FREEBUFFER Register Field Descriptions 1673

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13-63. RX1_FREEBUFFER Register Field Descriptions 1674
13-64. RX2_FREEBUFFER Register Field Descriptions 1675
13-65. RX3_FREEBUFFER Register Field Descriptions 1676
13-66. RX_TEARDOWN Register Field Descriptions 1677
13-67. RX4_FREEBUFFER Register Field Descriptions 1678
13-68. RX5_FREEBUFFER Register Field Descriptions 1679
13-69. RX6_FREEBUFFER Register Field Descriptions 1680
13-70. RX7_FREEBUFFER Register Field Descriptions 1681
13-71. TX0_HDP Register Field Descriptions 1682
13-72. TX1_HDP Register Field Descriptions 1683
13-73. TX2_HDP Register Field Descriptions 1684
13-74. TX3_HDP Register Field Descriptions 1685
13-75. TX4_HDP Register Field Descriptions 1686
13-76. TX5_HDP Register Field Descriptions 1687
13-77. TX6_HDP Register Field Descriptions 1688
13-78. TX7_HDP Register Field Descriptions 1689
13-79. RX0_HDP Register Field Descriptions 1690
13-80. RX1_HDP Register Field Descriptions 1691
13-81. RX2_HDP Register Field Descriptions 1692
13-82. RX3_HDP Register Field Descriptions 1693
13-83. RX4_HDP Register Field Descriptions 1694
13-84. RX5_HDP Register Field Descriptions 1695
13-85. RX6_HDP Register Field Descriptions 1696
13-86. RX7_HDP Register Field Descriptions 1697
13-87. TX0_CP Register Field Descriptions 1698
13-88. TX1_CP Register Field Descriptions 1699
13-89. TX2_CP Register Field Descriptions 1700
13-90. TX3_CP Register Field Descriptions 1701
13-91. TX4_CP Register Field Descriptions 1702
13-92. TX5_CP Register Field Descriptions 1703
13-93. TX6_CP Register Field Descriptions 1704
13-94. TX7_CP Register Field Descriptions 1705
13-95. RX0_CP Register Field Descriptions 1706
13-96. RX1_CP Register Field Descriptions 1707
13-97. RX2_CP Register Field Descriptions 1708
13-98. RX3_CP Register Field Descriptions 1709
13-99. RX4_CP Register Field Descriptions 1710
13-100. RX5_CP Register Field Descriptions 1711
13-101. RX6_CP Register Field Descriptions 1712
13-102. RX7_CP Register Field Descriptions 1713
13-103. CPDMA_SOFT_RESET Register Field Descriptions 1714
13-104. DMACONTROL Register Field Descriptions 1715
13-105. DMASTATUS Register Field Descriptions 1717
13-106. TX_CONTROL Register Field Descriptions 1718
13-107. RX_BUFFER_OFFSET Register Field Descriptions 1719
13-108. EMCONTROL Register Field Descriptions 1720
13-109. TX_PRI0_RATE Register Field Descriptions 1721
13-110. TX_PRI1_RATE Register Field Descriptions 1722
13-111. TX_PRI2_RATE Register Field Descriptions 1723

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13-112. TX_PRI3_RATE Register Field Descriptions 1724
13-113. TX_PRI4_RATE Register Field Descriptions 1725
13-114. TX_PRI5_RATE Register Field Descriptions 1726
13-115. TX_PRI6_RATE Register Field Descriptions 1727
13-116. TX_PRI7_RATE Register Field Descriptions 1728
13-117. TX_TEARDOWN Register Field Descriptions 1729
13-118. CPSW_CPTS REGISTERS 1729
13-119. CPTS_IDVER Register Field Descriptions 1731
13-120. CPTS_TS_PUSH Register Field Descriptions 1732
13-121. CPTS_TS_LOAD_VAL Register Field Descriptions 1733
13-122. CPTS_TS_LOAD_EN Register Field Descriptions 1734
13-123. CPTS_INTSTAT_RAW Register Field Descriptions 1735
13-124. CPTS_INTSTAT_MASKED Register Field Descriptions 1736
13-125. CPTS_CONTROL Register Field Descriptions 1737
13-126. CPTS_INT_ENABLE Register Field Descriptions 1738
13-127. CPTS_EVENT_POP Register Field Descriptions 1739
13-128. CPTS_EVENT_LOW Register Field Descriptions 1740
13-129. CPTS_EVENT_HIGH Register Field Descriptions 1741
13-130. CPTS_RFTCLK_SEL Register Field Descriptions 1742
13-131. CPSW_PORT REGISTERS 1742
13-132. P0_CONTROL Register Field Descriptions 1745
13-133. P0_BLK_CNT Register Field Descriptions 1746
13-134. P0_TX_IN_CTL Register Field Descriptions 1747
13-135. P0_PORT_VLAN Register Field Descriptions 1748
13-136. P0_TX_PRI_MAP Register Field Descriptions 1749
13-137. P1_CONTROL Register Field Descriptions 1750
13-138. P1_MAX_BLKS Register Field Descriptions 1752
13-139. P1_BLK_CNT Register Field Descriptions 1753
13-140. P1_TX_IN_CTL Register Field Descriptions 1754
13-141. P1_PORT_VLAN Register Field Descriptions 1755
13-142. P0_CPDMA_TX_PRI_MAP Register Field Descriptions 1756
13-143. P1_TX_PRI_MAP Register Field Descriptions 1757
13-144. P1_TS_SEQ_MTYPE Register Field Descriptions 1758
13-145. P1_SA_LO Register Field Descriptions 1759
13-146. P1_SA_HI Register Field Descriptions 1760
13-147. P1_SEND_PERCENT Register Field Descriptions 1761
13-148. P1_RX_DSCP_PRI_MAP0 Register Field Descriptions 1762
13-149. P1_RX_DSCP_PRI_MAP1 Register Field Descriptions 1763
13-150. P1_RX_DSCP_PRI_MAP2 Register Field Descriptions 1764
13-151. P1_RX_DSCP_PRI_MAP3 Register Field Descriptions 1765
13-152. P0_CPDMA_RX_CH_MAP Register Field Descriptions 1766
13-153. P1_RX_DSCP_PRI_MAP4 Register Field Descriptions 1767
13-154. P1_RX_DSCP_PRI_MAP5 Register Field Descriptions 1768
13-155. P1_RX_DSCP_PRI_MAP6 Register Field Descriptions 1769
13-156. P1_RX_DSCP_PRI_MAP7 Register Field Descriptions 1770
13-157. P0_RX_DSCP_PRI_MAP0 Register Field Descriptions 1771
13-158. P2_CONTROL Register Field Descriptions 1772
13-159. P0_RX_DSCP_PRI_MAP1 Register Field Descriptions 1774
13-160. P2_MAX_BLKS Register Field Descriptions 1775

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13-161. P2_BLK_CNT Register Field Descriptions 1776
13-162. P2_TX_IN_CTL Register Field Descriptions 1777
13-163. P2_PORT_VLAN Register Field Descriptions 1778
13-164. P2_TX_PRI_MAP Register Field Descriptions 1779
13-165. P2_TS_SEQ_MTYPE Register Field Descriptions 1780
13-166. P2_SA_LO Register Field Descriptions 1781
13-167. P2_SA_HI Register Field Descriptions 1782
13-168. P2_SEND_PERCENT Register Field Descriptions 1783
13-169. P0_RX_DSCP_PRI_MAP2 Register Field Descriptions 1784
13-170. P2_RX_DSCP_PRI_MAP0 Register Field Descriptions 1785
13-171. P2_RX_DSCP_PRI_MAP1 Register Field Descriptions 1786
13-172. P2_RX_DSCP_PRI_MAP2 Register Field Descriptions 1787
13-173. P2_RX_DSCP_PRI_MAP3 Register Field Descriptions 1788
13-174. P2_RX_DSCP_PRI_MAP4 Register Field Descriptions 1789
13-175. P2_RX_DSCP_PRI_MAP5 Register Field Descriptions 1790
13-176. P2_RX_DSCP_PRI_MAP6 Register Field Descriptions 1791
13-177. P2_RX_DSCP_PRI_MAP7 Register Field Descriptions 1792
13-178. P0_RX_DSCP_PRI_MAP3 Register Field Descriptions 1793
13-179. P0_RX_DSCP_PRI_MAP4 Register Field Descriptions 1794
13-180. P0_RX_DSCP_PRI_MAP5 Register Field Descriptions 1795
13-181. P0_RX_DSCP_PRI_MAP6 Register Field Descriptions 1796
13-182. P0_RX_DSCP_PRI_MAP7 Register Field Descriptions 1797
13-183. P0_MAX_BLKS Register Field Descriptions 1798
13-184. CPSW_SL REGISTERS 1798
13-185. IDVER Register Field Descriptions 1800
13-186. SOFT_RESET Register Field Descriptions 1801
13-187. RX_MAXLEN Register Field Descriptions 1802
13-188. BOFFTEST Register Field Descriptions 1803
13-189. RX_PAUSE Register Field Descriptions 1804
13-190. TX_PAUSE Register Field Descriptions 1805
13-191. EMCONTROL Register Field Descriptions 1806
13-192. RX_PRI_MAP Register Field Descriptions 1807
13-193. MACCONTROL Register Field Descriptions 1808
13-194. TX_GAP Register Field Descriptions 1810
13-195. MACSTATUS Register Field Descriptions 1811
13-196. CPSW_SS REGISTERS 1811
13-197. ID_VER Register Field Descriptions 1813
13-198. STAT_PORT_EN Register Field Descriptions 1814
13-199. PTYPE Register Field Descriptions 1815
13-200. SOFT_IDLE Register Field Descriptions 1816
13-201. THRU_RATE Register Field Descriptions 1817
13-202. GAP_THRESH Register Field Descriptions 1818
13-203. TX_START_WDS Register Field Descriptions 1819
13-204. FLOW_CONTROL Register Field Descriptions 1820
13-205. CONTROL Register Field Descriptions 1821
13-206. VLAN_LTYPE Register Field Descriptions 1822
13-207. TS_LTYPE Register Field Descriptions 1823
13-208. DLR_LTYPE Register Field Descriptions 1824
13-209. SOFT_RESET Register Field Descriptions 1825

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13-210. CPSW_WR REGISTERS 1825
13-211. IDVER Register Field Descriptions 1827
13-212. C2_RX_STAT Register Field Descriptions 1828
13-213. C2_TX_STAT Register Field Descriptions 1829
13-214. C2_MISC_STAT Register Field Descriptions 1830
13-215. C0_RX_IMAX Register Field Descriptions 1831
13-216. C0_TX_IMAX Register Field Descriptions 1832
13-217. INT_CONTROL Register Field Descriptions 1833
13-218. C1_RX_IMAX Register Field Descriptions 1834
13-219. C1_TX_IMAX Register Field Descriptions 1835
13-220. C2_RX_IMAX Register Field Descriptions 1836
13-221. C2_TX_IMAX Register Field Descriptions 1837
13-222. RGMII_CTL Register Field Descriptions 1838
13-223. C0_RX_THRESH_EN Register Field Descriptions 1839
13-224. C0_RX_EN Register Field Descriptions 1840
13-225. C0_TX_EN Register Field Descriptions 1841
13-226. C0_MISC_EN Register Field Descriptions 1842
13-227. C1_RX_THRESH_EN Register Field Descriptions 1843
13-228. C1_RX_EN Register Field Descriptions 1844
13-229. SOFT_RESET Register Field Descriptions 1845
13-230. C1_TX_EN Register Field Descriptions 1846
13-231. C1_MISC_EN Register Field Descriptions 1847
13-232. C2_RX_THRESH_EN Register Field Descriptions 1848
13-233. C2_RX_EN Register Field Descriptions 1849
13-234. C2_TX_EN Register Field Descriptions 1850
13-235. C2_MISC_EN Register Field Descriptions 1851
13-236. C0_RX_THRESH_STAT Register Field Descriptions 1852
13-237. C0_RX_STAT Register Field Descriptions 1853
13-238. C0_TX_STAT Register Field Descriptions 1854
13-239. C0_MISC_STAT Register Field Descriptions 1855
13-240. CONTROL Register Field Descriptions 1856
13-241. C1_RX_THRESH_STAT Register Field Descriptions 1857
13-242. C1_RX_STAT Register Field Descriptions 1858
13-243. C1_TX_STAT Register Field Descriptions 1859
13-244. C1_MISC_STAT Register Field Descriptions 1860
13-245. C2_RX_THRESH_STAT Register Field Descriptions 1861
14-1. Unsupported Features 1864
14-2. PWMSS Connectivity Attributes 1866
14-3. PWMSS Clock Signals 1866
14-4. PWMSS Pin List 1866
14-5. ePWM Module Control and Status Registers Grouped by Submodule 1872
14-6. Submodule Configuration Parameters 1874
14-7. Time-Base Submodule Registers 1879
14-8. Key Time-Base Signals 1880
14-9. Counter-Compare Submodule Registers 1888
14-10. Counter-Compare Submodule Key Signals 1888
14-11. Action-Qualifier Submodule Registers 1892
14-12. Action-Qualifier Submodule Possible Input Events 1893
14-13. Action-Qualifier Event Priority for Up-Down-Count Mode 1895

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14-14. Action-Qualifier Event Priority for Up-Count Mode 1895
14-15. Action-Qualifier Event Priority for Down-Count Mode 1895
14-16. Behavior if CMPA/CMPB is Greater than the Period 1896
14-17. EPWMx Initialization for 1899
14-18. EPWMx Run Time Changes for 1899
14-19. EPWMx Initialization for 1901
14-20. EPWMx Run Time Changes for 1901
14-21. EPWMx Initialization for 1903
14-22. EPWMx Run Time Changes for 1903
14-23. EPWMx Initialization for 1905
14-24. EPWMx Run Time Changes for 1905
14-25. EPWMx Initialization for 1907
14-26. EPWMx Run Time Changes for 1907
14-27. EPWMx Initialization for 1909
14-28. EPWMx Run Time Changes for 1909
14-29. Dead-Band Generator Submodule Registers 1910
14-30. Classical Dead-Band Operating Modes 1912
14-31. PWM-Chopper Submodule Registers 1914
14-32. Trip-Zone Submodule Registers 1919
14-33. Possible Actions On a Trip Event 1920
14-34. Event-Trigger Submodule Registers 1922
14-35. Resolution for PWM and HRPWM 1927
14-36. HRPWM Submodule Registers 1928
14-37. Relationship Between MEP Steps, PWM Frequency and Resolution 1929
14-38. CMPA vs Duty (left), and [CMPA:CMPAHR] vs Duty (right) 1930
14-39. SFO Library Version Comparison 1933
14-40. SFO Library Routines 1934
14-41. Factor Values 1935
14-42. SFO V5 Library Routines 1938
14-43. Software Functions 1941
14-44. EPWM1 Initialization for 1948
14-45. EPWM2 Initialization for 1948
14-46. EPWM3 Initialization for 1948
14-47. EPWM1 Initialization for 1951
14-48. EPWM2 Initialization for 1951
14-49. EPWM1 Initialization for 1954
14-50. EPWM2 Initialization for 1954
14-51. EPWM1 Initialization for 1957
14-52. EPWM2 Initialization for 1957
14-53. EPWM3 Initialization for 1958
14-54. EPWM1 Initialization for 1963
14-55. EPWM2 Initialization for 1963
14-56. EPWM3 Initialization for 1964
14-57. EPWM1 Initialization for 1967
14-58. EPWM2 Initialization for 1967
14-59. ECAP Initialization for CAP Mode Absolute Time, Rising Edge Trigger 1982
14-60. ECAP Initialization for CAP Mode Absolute Time, Rising and Falling Edge Trigger 1984
14-61. ECAP Initialization for CAP Mode Delta Time, Rising Edge Trigger 1986
14-62. ECAP Initialization for CAP Mode Delta Time, Rising and Falling Edge Triggers 1988

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14-63. ECAP Initialization for APWM Mode 1990
14-64. ECAP1 Initialization for Multichannel PWM Generation with Synchronization 1992
14-65. ECAP2 Initialization for Multichannel PWM Generation with Synchronization 1992
14-66. ECAP3 Initialization for Multichannel PWM Generation with Synchronization 1992
14-67. ECAP4 Initialization for Multichannel PWM Generation with Synchronization 1992
14-68. ECAP1 Initialization for Multichannel PWM Generation with Phase Control 1995
14-69. ECAP2 Initialization for Multichannel PWM Generation with Phase Control 1995
14-70. ECAP3 Initialization for Multichannel PWM Generation with Phase Control 1995
14-71. Control and Status Register Set 1996
14-72. Time-Stamp Counter Register (TSCTR) Field Descriptions 1996
14-73. Counter Phase Control Register (CTRPHS) Field Descriptions 1997
14-74. Capture 1 Register (CAP1) Field Descriptions 1997
14-75. Capture 2 Register (CAP2) Field Descriptions 1998
14-76. Capture 3 Register (CAP3) Field Descriptions 1998
14-77. Capture 4 Register (CAP4) Field Descriptions 1999
14-78. ECAP Control Register 1 (ECCTL1) Field Descriptions 1999
14-79. ECAP Control Register 2 (ECCTL2) Field Descriptions 2001
14-80. ECAP Interrupt Enable Register (ECEINT) Field Descriptions 2003
14-81. ECAP Interrupt Flag Register (ECFLG) Field Descriptions 2004
14-82. ECAP Interrupt Clear Register (ECCLR) Field Descriptions 2005
14-83. ECAP Interrupt Forcing Register (ECFRC) Field Descriptions 2006
14-84. Revision ID Register (REVID) Field Descriptions 2007
14-85. Quadrature Decoder Truth Table 2014
14-86. eQEP Registers 2029
14-87. eQEP Position Counter Register (QPOSCNT) Field Descriptions 2030
14-88. eQEP Position Counter Initialization Register (QPOSINIT) Field Descriptions 2030
14-89. eQEP Maximum Position Count Register (QPOSMAX) Field Descriptions 2030
14-90. eQEP Position-Compare Register (QPOSCMP) Field Descriptions 2031
14-91. eQEP Index Position Latch Register (QPOSILAT) Field Descriptions 2031
14-92. eQEP Strobe Position Latch Register (QPOSSLAT) Field Descriptions 2031
14-93. eQEP Position Counter Latch Register (QPOSLAT) Field Descriptions 2032
14-94. eQEP Unit Timer Register (QUTMR) Field Descriptions 2032
14-95. eQEP Unit Period Register (QUPRD) Field Descriptions 2032
14-96. eQEP Watchdog Timer Register (QWDTMR) Field Descriptions 2033
14-97. eQEP Watchdog Period Register (QWDPRD) Field Description 2033
14-98. eQEP Decoder Control Register (QDECCTL) Field Descriptions 2034
14-99. eQEP Control Register (QEPCTL) Field Descriptions 2035
14-100. eQEP Capture Control Register (QCAPCTL) Field Descriptions 2037
14-101. eQEP Position-Compare Control Register (QPOSCTL) Field Descriptions 2038
14-102. eQEP Interrupt Enable Register (QEINT) Field Descriptions 2039
14-103. eQEP Interrupt Flag Register (QFLG) Field Descriptions 2040
14-104. eQEP Interrupt Clear Register (QCLR) Field Descriptions 2041
14-105. eQEP Interrupt Force Register (QFRC) Field Descriptions 2043
14-106. eQEP Status Register (QEPSTS) Field Descriptions 2044
14-107. eQEP Capture Time Register (QCTMR) Field Descriptions 2045
14-108. eQEP Capture Period Register (QCPRD) Field Descriptions 2045
14-109. eQEP Capture Timer Latch Register (QCTMRLAT) Field Descriptions 2045
14-110. eQEP Capture Period Latch Register (QCPRDLAT) Field Descriptions 2046
14-111. eQEP Revision ID Register (REVID) Field Descriptions 2046

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15-1. USB Connectivity Attributes 2051
15-2. USB Clock Signals 2051
15-3. USB Pin List 2052
15-4. PERI_TXCSR Register Bit Configuration for Bulk IN Transactions 2067
15-5. PERI_RXCSR Register Bit Configuration for Bulk OUT Transactions 2069
15-6. PERI_TXCSR Register Bit Configuration for Isochronous IN Transactions 2071
15-7. PERI_RXCSR Register Bit Configuration for Isochronous OUT Transactions 2072
15-8. Isochronous OUT Error Handling: Peripheral Mode 2073
15-9. Packet Descriptor Word 0 (PD0) Bit Field Descriptions 2092
15-10. Packet Descriptor Word 1 (PD1) Bit Field Descriptions 2093
15-11. Packet Descriptor Word 2 (PD2) Bit Field Descriptions 2093
15-12. Packet Descriptor Word 3 (PD3) Bit Field Descriptions 2093
15-13. Packet Descriptor Word 4 (PD4) Bit Field Descriptions 2094
15-14. Packet Descriptor Word 5 (PD5) Bit Field Descriptions 2094
15-15. Packet Descriptor Word 6 (PD6) Bit Field Descriptions 2094
15-16. Packet Descriptor Word 7 (PD7) Bit Field Descriptions 2094
15-17. Buffer Descriptor Word 0 (BD0) Bit Field Descriptions 2095
15-18. Buffer Descriptor Word 1 (BD1) Bit Field Descriptions 2095
15-19. Buffer Descriptor Word 2 (BD2) Bit Field Descriptions 2095
15-20. Buffer Descriptor Word 3 (BD3) Bit Field Descriptions 2095
15-21. Buffer Descriptor Word 4 (BD4) Bit Field Descriptions 2096
15-22. Buffer Descriptor Word 5 (BD5) Bit Field Descriptions 2096
15-23. Buffer Descriptor Word 6 (BD6) Bit Field Descriptions 2096
15-24. Buffer Descriptor Word 7 (BD7) Bit Field Descriptions 2096
15-25. Teardown Descriptor Word 0 Bit Field Descriptions 2097
15-26. Teardown Descriptor Words 1 to 7 Bit Field Descriptions 2097
15-27. Queue-Endpoint Assignments 2098
15-28. 53 Bytes Test Packet Content 2114
15-29. USBSS Registers 2116
15-30. USBSS Revision Register (REVREG) Field Description 2117
15-31. USBSS SYSCONFIG Register (SYSCONFIG) Field Descriptions 2118
15-32. USBSS End of Interrupt Register (EOI) Field Descriptions 2119
15-33. USBSS IRQ_STATUS_RAW (IRQSTATRAW) Field Descriptions 2120
15-34. USBSS IRQ_STATUS (IRQSTAT) Field Descriptions 2121
15-35. USBSS IRQ_ENABLE_SET Register (IRQENABLER) Field Descriptions 2122
15-36. USBSS IRQ_ENABLE_CLR Register (IRQCLEARR) Field Descriptions 2123
15-37. USBSS IRQ_DMA_THRESHOLD_TX0_0 Register (IRQDMATHOLDTX00) Field Descriptions 2124
15-38. USBSS IRQ_DMA_THRESHOLD_TX0_1 Register (IRQDMATHOLDTX01) Field Descriptions 2124
15-39. USBSS IRQ_DMA_THRESHOLD_TX0_2 Register (IRQDMATHOLDTX02) Field Descriptions 2125
15-40. USBSS IRQ_DMA_THRESHOLD_TX0_3 Register (IRQDMATHOLDTX03) Field Descriptions 2125
15-41. USBSS IRQ_DMA_THRESHOLD_RX0_0 Register (IRQDMATHOLDRX00) Field Descriptions 2126
15-42. USBSS IRQ_DMA_THRESHOLD_RX0_1 Register (IRQDMATHOLDRX00) Field Descriptions 2126
15-43. USBSS IRQ_DMA_THRESHOLD_RX0_2 Register (IRQDMATHOLDRX02) Field Descriptions 2127
15-44. USBSS IRQ_DMA_THRESHOLD_RX0_3 Register (IRQDMATHOLDRX03) Field Descriptions 2127
15-45. USBSS IRQ_DMA_THRESHOLD_TX1_0 Register (IRQDMATHOLDTX10) Field Descriptions 2128
15-46. USBSS IRQ_DMA_THRESHOLD_TX1_1 Register (IRQDMATHOLDTX11) Field Descriptions 2128
15-47. USBSS IRQ_DMA_THRESHOLD_TX1_2 Register (IRQDMATHOLDTX12) Field Descriptions 2129
15-48. USBSS IRQ_DMA_THRESHOLD_TX1_3 Register (IRQDMATHOLDTX13) Field Descriptions 2129
15-49. USBSS IRQ_DMA_THRESHOLD_RX1_0 Register (IRQDMATHOLDRX10) Field Descriptions 2130

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15-50. USBSS IRQ_DMA_THRESHOLD_RX1_1 Register (IRQDMATHOLDRX11) Field Descriptions 2130
15-51. USBSS IRQ_DMA_THRESHOLD_RX1_2 Register (IRQDMATHOLDRX12) Field Descriptions 2131
15-52. USBSS IRQ_DMA_THRESHOLD_RX1_3 Register (IRQDMATHOLDRX13) Field Descriptions 2131
15-53. USBSS IRQ_DMA_ENABLE_0 Register (IRQDMAENABLE0) Field Descriptions 2132
15-54. USBSS IRQ_DMA_ENABLE_1 Register (IRQDMAENABLE1) Field Descriptions 2134
15-55. USBSS IRQ_FRAME_THRESHOLD_TX0_0 Register (IRQFRAMETHOLD00) Field Descriptions 2136
15-56. USBSS IRQ_FRAME_THRESHOLD_TX0_1 Register (IRQFRAMETHOLDTX01) Field Descriptions 2136
15-57. USBSS IRQ_FRAME_THRESHOLD_TX0_2 Register (IRQFRAMETHOLDTX02) Field Descriptions 2137
15-58. USBSS IRQ_FRAME_THRESHOLD_TX0_3 Register (IRQFRAMETHOLDTX03) Field Descriptions 2137
15-59. USBSS IRQ_FRAME_THRESHOLD_RX0_0 Register (IRQFRAMETHOLDRX00) Field Descriptions 2138
15-60. USBSS IRQ_FRAME_THRESHOLD_RX0_1 Register (IRQFRAMETHOLDRX01) Field Descriptions 2138
15-61. USBSS IRQ_FRAME_THRESHOLD_RX0_2 Register (IRQFRAMETHOLDRX02) Field Descriptions 2139
15-62. USBSS IRQ_FRAME_THRESHOLD_RX0_3 Register (IRQFRAMETHOLDRX03) Field Descriptions 2139
15-63. USBSS IRQ_FRAME_THRESHOLD_TX1_0 Register (IRQFRAMETHOLD10) Field Descriptions 2140
15-64. USBSS IRQ_FRAME_THRESHOLD_TX1_1 Register (IRQFRAMETHOLDTX11) Field Descriptions 2140
15-65. USBSS IRQ_FRAME_THRESHOLD_TX1_2 Register (IRQFRAMETHOLDTX12) Field Descriptions 2141
15-66. USBSS IRQ_FRAME_THRESHOLD_TX1_3 Register (IRQFRAMETHOLDTX13) Field Descriptions 2141
15-67. USBSS IRQ_FRAME_THRESHOLD_RX1_0 Register (IRQFRAMETHOLDRX10) Field Descriptions 2142
15-68. USBSS IRQ_FRAME_THRESHOLD_RX1_1 Register (IRQFRAMETHOLDRX11) Field Descriptions 2142
15-69. USBSS IRQ_FRAME_THRESHOLD_RX1_2 Register (IRQFRAMETHOLDRX12) Field Descriptions 2143
15-70. USBSS IRQ_FRAME_THRESHOLD_RX1_3 Register (IRQFRAMETHOLDRX13) Field Descriptions 2143
15-71. USBSS IRQ_FRAME_ENABLE_0 Register (IRQFRAMEENABLE0) Field Descriptions 2144
15-72. USBSS IRQ_FRAME_ENABLE_1 Register (IRQFRAMEENABLE1) Field Descriptions 2146
15-73. USB0 Controller Registers 2148
15-74. USB0 Revision Register (USB0REV) Field Descriptions 2149
15-75. USB0 Control Register (USB0CTRL) Field Descriptions..... 2150
15-76. USB0 Status Register (USB0STAT) Field Descriptions..... 2151
15-77. USB0 IRQ_MERGED_STATUS Register (USB0IRQMSTAT) Field Descriptions 2151
15-78. USB0 IRQ_EOI Register (USB0IRQEOI) Field Descriptions 2152
15-79. USB0 IRQ_STATUS_RAW_0 Register (USB0IRQSTATRAW0) Field Descriptions 2153
15-80. USB0 IRQ_STATUS_RAW_1 Register (USB0IRQSTATRAW1) Field Descriptions 2155
15-81. USB0 IRQ_STATUS_0 Register (USB0IRQSTAT0) Field Descriptions 2157
15-82. USB0 IRQ_STATUS_1 Register (USB0IRQSTAT1) Field Descriptions 2159
15-83. USB0 IRQ_ENABLE_SET_0 Register (USB0IRQENABLESET0) Field Descriptions 2161
15-84. USB0 IRQ_ENABLE_SET_1 Register (USB0IRQENABLESET1) Field Descriptions 2163
15-85. USB0 IRQ_ENABLE_CLR_0 Register (USB0IRQENABLECLR0) Field Descriptions 2165
15-86. USB0 IRQ_ENABLE_CLR_1 Register (USB0IRQENABLECLR1) Field Descriptions 2167
15-87. USB0 Tx Mode Register (USB0TXMODE) Field Descriptions 2169
15-88. USB0 Rx Mode Register (USB0RXMODE) Field Descriptions 2171
15-89. USB0 Generic RNDIS EPn Size Register (USB0GENRNDISEPn) Field Descriptions 2173
15-90. USB0 Auto Req Register (USB0AUTOREQ) Field Descriptions..... 2174
15-91. USB0 SRP Fix Time Register (USB0SRPFIXTIME) Field Descriptions 2177
15-92. USB0 Teardown Register (USB0TDOWN) Field Descriptions 2177
15-93. USB0 PHY UTMI Register (USB0UTMI) Field Descriptions..... 2178
15-94. USB0 MGC UTMI Loopback Register (USB0UTMILB) Field Descriptions 2179
15-95. USB0 Mode Register (USB0MODE) Field Descriptions 2180
15-96. USB1 Controller Registers 2181
15-97. USB1 Revision Register (USB1REV) Field Descriptions 2182
15-98. USB1 Control Register (USB1CTRL) Field Descriptions 2183

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15-99. USB1 Status Register (USB1STAT) Field Descriptions 2184
15-100. USB1 IRQ_MERGED_STATUS Register (USB1IRQMSTAT) Field Descriptions 2184
15-101. USB1 IRQ_EOI Register (USB1IRQEOI) Field Descriptions 2185
15-102. USB1 IRQ_STATUS_RAW_0 Register (USB1IRQSTATRAW1) Field Descriptions 2186
15-103. USB1 IRQ_STATUS_0 Register (USBIRQSTAT0) Field Descriptions 2188
15-104. USB1 IRQ_STATUS_1 Register (USB1IRQSTAT1) Field Descriptions 2190
15-105. USB1 IRQ_ENABLE_SET_0 Register (USB1IRQENABLESET0) Field Descriptions 2192
15-106. USB1 IRQ_ENABLE_SET_1 Register (USB1IRQENABLESET1) Field Descriptions 2194
15-107. USB1 IRQ_ENABLE_CLR_0 Register (USB1IRQENABLECLR0) Field Descriptions 2196
15-108. USB1 IRQ_ENABLE_CLR_1 Register (USB1IREENABLECLR1) Field Descriptions 2198
15-109. USB1 Tx Mode Register (USB1TXMODE) Field Descriptions 2200
15-110. USB1 Rx Mode Register (USB1RXMODE) Field Descriptions 2202
15-111. USB1 Generic RNDIS EP N Size Register (USB1GENRNDISEPn) Field Descriptions 2204
15-112. USB1 Auto Req Register (USB1AUTOREQ) Field Descriptions 2205
15-113. USB1 SRP Fix Time Register (USB1SRPFIXTIME) Field Descriptions 2208
15-114. USB1 Teardown Register (USB1TDOWN) Field Descriptions 2208
15-115. USB1 PHY UTMI Register (USB1UTMI) Field Descriptions 2209
15-116. USB1 MGC UTMI Loopback Register (USB1UTMILB) Field Descriptions 2210
15-117. USB1 Mode Register (USB1MODE) Field Descriptions 2211
15-118. CPPI DMA Controller Registers 2212
15-119. CPPI DMA Revision Registe (DMAREVID) Field Descriptions 2213
15-120. CPPI DMA Teardown Free Descriptor Queue Control Register (TDFDQ) Field Descriptions 2213
15-121. CPPI DMA Emulation Control Register (DMAEMU) Field Descriptions 2214
15-122. CPPI Mem1 Base Address Register (DMAMEM1BA) Field Descriptions 2214
15-123. CPPI Mem1 Mask Address Register (DMAMEM1MASK) Field Descriptions 2215
15-124. Tx Channel N Global Configuration Register (TXGCRn) Field Descriptions 2215
15-125. Rx Channel N Global Configuration Register (RXGCRn) Field Descriptions 2216
15-126. Rx Channel N Host Packet Configuration Register A (RXHPCRAn) Field Descriptions 2217
15-127. Rx Channel N Host Packet Configuration Register B (RXHPCRBn) Field Descriptions 2218
15-128. CPPI DMA Scheduler Registers 2219
15-129. CPPI DMA Scheduler Control Register (DMA_SCHED_CTRL) Field Descriptions 2219
15-130. CPPI DMA Scheduler Table Word N (WORDn) Field Descriptions 2220
15-131. CPPI DMA Queue Manager Registers 2222
15-132. Queue Manager Revision Register (QMGRREVID) Field Descriptions 2223
15-133. Queue Manager Reset Register (QMGRRST) Field Descriptions 2223
15-134. Queue Manager Queue Diversion Register (DIVERSION) Field Descriptions 2224
15-135. Queue Manager Free Descriptor/Buffer Starvation Count Register 0 (FDBSC0) Field Descriptions 2225
15-136. Queue Manager Free Descriptor/Buffer Starvation Count Register 1 (FDBSC1) Field Descriptions 2225
15-137. Queue Manager Free Descriptor/Buffer Starvation Count Register 2 (FDBSC2) Field Descriptions 2227
15-138. Queue Manager Free Descriptor/Buffer Starvation Count Register 3 (FDBSC3) Field Descriptions 2227
15-139. Queue Manager Free Descriptor/Buffer Starvation Count Register 4 (FDBSC4) Field Descriptions 2229
15-140. Queue Manager Free Descriptor/Buffer Starvation Count Register 5 (FDBSC5) Field Descriptions 2229
15-141. Queue Manager Free Descriptor/Buffer Starvation Count Register 6 (FDBSC6) Field Descriptions 2231
15-142. Queue Manager Free Descriptor/Buffer Starvation Count Register 7 (FDBSC7) Field Descriptions 2231
15-143. Queue Manager Linking RAM Region 0 Base Address Register (LRAM0BASE) Field Descriptions 2233
15-144. Queue Manager Linking RAM Region 0 Size Register (LRAM0SIZE) Field Descriptions 2233
15-145. Queue Manager Linking RAM Region 1 Base Address Register (LRAM1BASE) Field Descriptions 2234
15-146. Queue Manager Queue Pending Register 0 (PEND0) Field Descriptions 2234
15-147. Queue Manager Queue Pending Register 1 (PEND1) Field Descriptions 2235

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15-148. Queue Manager Queue Pending Register 2 (PEND1) Field Descriptions 2235
15-149. Queue Manager Queue Pending Register 3 (PEND3) Field Descriptions 2235
15-150. Queue Manager Queue Pending Register 4 (PEND4) Field Descriptions 2236
15-151. Queue Manager Memory Region R Base Address Register (QMEMRBASEr) Field Descriptions 2236
15-152. Queue Manager Memory Region R Control Register (QMEMRCTRLr) Field Descriptions 2237
15-153. Queue Manager Queue N Register A (CTRLAn) Field Descriptions 2238
15-154. Queue Manager Queue N Register B (CTRLBn) Field Descriptions 2238
15-155. Queue Manager Queue N Register C (CTRLCn) Field Descriptions 2239
15-156. Queue Manager Queue N Register D (CTRLDn) Field Descriptions 2239
15-157. Queue Manager Queue N Status Register A (QSTATAn) Field Descriptions 2240
15-158. Queue Manager Queue N Status Register B (QSTATBn) Field Descriptions 2240
15-159. Queue Manager Queue N Status Register C (QSTATCn) Field Descriptions 2241
15-160. Common USB Registers 2242
15-161. Function Address Register (USBn_FADDR) Field Descriptions 2243
15-162. Power Management Register (USBn_POWER) Field Descriptions 2243
15-163. Interrupt Register for Endpoint 0 Plus Transmit Endpoints 1 to 15 (USBn_INTRTX) Field Descriptions . 2244
15-164. Interrupt Register for Receive Endpoints 1 to 15 (INTRRX) Field Descriptions 2245
15-165. Interrupt Enable Register for INTRTX (INTRTXE) Field Descriptions 2246
15-166. Interrupt Enable Register for INTRRX (INTRRXE) Field Descriptions 2247
15-167. Interrupt Register for Common USB Interrupts (USBn_INTRUSB) Field Descriptions 2248
15-168. Interrupt Enable Register for INTRUSB (USBn_INTRUSBE) Field Descriptions 2248
15-169. Frame Number Register (USBn_FRAME) Field Descriptions 2249
15-170. Index Register for Selecting the Endpoint Status and Control Registers (USBn_INDEX) Field Descriptions 2249
15-171. Register to Enable the USB 2.0 Test Modes (USBn_TESTMODE) Field Descriptions 2250
15-172. Indexed Region Registers 2251
15-173. Maximum Packet Size for Peripheral/Host Transmit Endpoint (USBn_TXMAXP) Field Descriptions 2251
15-174. Control Status Register for Endpoint 0 in Peripheral Mode (USBn_PERI_CSR0) Field Descriptions 2252
15-175. Control Status Register for Endpoint 0 in Host Mode (USBn_HOST_CSR0) Field Descriptions 2253
15-176. Control Status Register for Peripheral Transmit Endpoint (USBn_PERI_TXCSR) Field Descriptions 2254
15-177. Control Status Register for Host Transmit Endpoint (USBn_HOST_TXCSR) Field Descriptions 2255
15-178. Maximum Packet Size for Peripheral Host Receive Endpoint (USBn_RXMAXP) Field Descriptions 2256
15-179. Control Status Register for Peripheral Receive Endpoint (USBn_PERI_RXCSR) Field Descriptions 2257
15-180. Control Status Register for Host Receive Endpoint (USBn_HOST_RXCSR) Field Descriptions 2259
15-181. Count 0 Register (USBn_COUNT0) Field Descriptions 2260
15-182. Receive Count Register (USBn_RXCOUNT) Field Descriptions 2261
15-183. Type Register (Host Mode Only) (USBn_HOST_TYPE0) Field Descriptions 2261
15-184. Transmit Type Register (Host Mode Only) (USBn_HOST_TXTYPE) Field Descriptions 2262
15-185. NAKLimit0 Register (Host Mode Only) (USBn_HOST_NAKLIMIT0) Field Descriptions 2263
15-186. Transmit Interval Register (Host Mode Only) (USBn_HOST_TXINTERVAL) Field Descriptions 2264
15-187. Receive Type Register (Host Mode Only) (USBn_HOST_RXTYPE) Field Descriptions 2265
15-188. Transmit Interval Register (Host Mode Only) (USBn_HOST_RXINTERVAL) Field Descriptions 2266
15-189. Configuration Data Register (USBn_CONFIGDATA) Field Descriptions 2267
15-190. FIFOs: Transmit and Receive FIFO Register for Endpoint 0 - 15 (USBn_FIFO0 – USBn_FIFO15) Field Descriptions 2268
15-191. Additional Control and Configuration Registers 2269
15-192. Device Control Register (USBn_DEVCTL) Field Descriptions 2269
15-193. Transmit Endpoint FIFO Size Register (USBn_TXFIFOSZ) Field Descriptions 2270
15-194. Receive Endpoint FIFO Size Register (USBn_RXFIFOSZ) Field Descriptions 2271

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15-195. Transmit Endpoint FIFO Address Register (USBn_TXFIFOADDR) Field Descriptions 2271
15-196. Receive Endpoint FIFO Address Register (USBn_RXFIFOADDR) Field Descriptions 2272
15-197. Hardware Version Register (USBn_HWVERS) Field Descriptions 2272
15-198. Target Endpoint Control Registers 2273
15-199. Transmit Function Address (USBn_TXFUNCADDRm) Field Descriptions 2273
15-200. Transmit Hub Address Register (USBn_TXHUBADDRm) Field Descriptions 2274
15-201. Transmit Hub Port Register (USBn_TXHUBPORTm) Field Descriptions 2274
15-202. Receive Function Address Register (USBn_RXFUNCADDRm) Field Descriptions 2275
15-203. Receive Hub Address Register (USBn_RXHUBADDRm) Field Descriptions 2275
15-204. Receive Hub Port Register (USBn_RXHUBPORTm) Field Descriptions 2277
15-205. Peripheral Mode Configuration and Status Register Mapping where n=0,1 (USB0 or USB1) and m=1-15 (endpoint number) 2278
15-206. Host Mode Configuration and Status Register Mapping where n=0,1 (USB0 or USB1) and m=1-15 (endpoint number) 2278
16-1. Mailbox Connectivity Attributes 2281
16-2. Mailbox Clock Signals 2281
16-3. SPINLOCK REGISTERS 2282
16-4. REV Register Field Descriptions 2286
16-5. SYSCONFIG Register Field Descriptions 2287
16-6. SYSTATUS Register Field Descriptions 2288
16-7. LOCK_REG_0 Register Field Descriptions 2289
16-8. LOCK_REG_1 Register Field Descriptions 2290
16-9. LOCK_REG_2 Register Field Descriptions 2291
16-10. LOCK_REG_3 Register Field Descriptions 2292
16-11. LOCK_REG_4 Register Field Descriptions 2293
16-12. LOCK_REG_5 Register Field Descriptions 2294
16-13. LOCK_REG_6 Register Field Descriptions 2295
16-14. LOCK_REG_7 Register Field Descriptions 2296
16-15. LOCK_REG_8 Register Field Descriptions 2297
16-16. LOCK_REG_9 Register Field Descriptions 2298
16-17. LOCK_REG_10 Register Field Descriptions 2299
16-18. LOCK_REG_11 Register Field Descriptions 2300
16-19. LOCK_REG_12 Register Field Descriptions 2301
16-20. LOCK_REG_13 Register Field Descriptions 2302
16-21. LOCK_REG_14 Register Field Descriptions 2303
16-22. LOCK_REG_15 Register Field Descriptions 2304
16-23. LOCK_REG_16 Register Field Descriptions 2305
16-24. LOCK_REG_17 Register Field Descriptions 2306
16-25. LOCK_REG_18 Register Field Descriptions 2307
16-26. LOCK_REG_19 Register Field Descriptions 2308
16-27. LOCK_REG_20 Register Field Descriptions 2309
16-28. LOCK_REG_21 Register Field Descriptions 2310
16-29. LOCK_REG_22 Register Field Descriptions 2311
16-30. LOCK_REG_23 Register Field Descriptions 2312
16-31. LOCK_REG_24 Register Field Descriptions 2313
16-32. LOCK_REG_25 Register Field Descriptions 2314
16-33. LOCK_REG_26 Register Field Descriptions 2315
16-34. LOCK_REG_27 Register Field Descriptions 2316
16-35. LOCK_REG_28 Register Field Descriptions 2317

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16-36. LOCK_REG_29 Register Field Descriptions 2318
16-37. LOCK_REG_30 Register Field Descriptions 2319
16-38. LOCK_REG_31 Register Field Descriptions 2320
17-1. Unsupported MMCHS Features 2322
17-2. MMCHS Connectivity Attributes 2323
17-3. MMCHS Clock Signals 2323
17-4. MMCHS Pin List 2324
17-5. DAT Line Direction for Data Transfer Modes 2324
17-6. ADPDATDIROQ and ADPDATDIRLS Signal States 2324
17-7. MMC/SD/SDIO Controller Pins and Descriptions 2327
17-8. Response Type Summary 2331
17-9. Local Power Management Features 2336
17-10. Clock Activity Settings 2336
17-11. Events 2337
17-12. Memory Size, BLEN, and Buffer Relationship 2344
17-13. MMC, SD, SDIO Responses in the SD_RSPxx Registers 2345
17-14. CC and TC Values Upon Error Detected 2346
17-15. MMC/SD/SDIO Controller Transfer Stop Command Summary 2353
17-16. MMC/SD/SDIO Hardware Status Features 2358
17-17. Global Init for Surrounding Modules 2359
17-18. MMC/SD/SDIO Controller Wake-Up Configuration 2360
17-19. MMC/SD/SDIO Registers 2364
17-20. System Configuration Register (SD_SYSCONFIG) Field Descriptions 2366
17-21. System Status Register (SD_SYSSTATUS) Field Descriptions 2368
17-22. Card Status Response Error (SD_CSRE) Field Descriptions 2368
17-23. System Test Register (SD_SYSTEST) Field Descriptions 2369
17-24. Configuration Register (SD_CON) Field Descriptions 2372
17-25. Power Counter Register (SD_PWCNT) Field Descriptions 2375
17-26. Card Status Response Error (SD_SDMASA) Field Descriptions 2375
17-27. Transfer Length Configuration Register (SD_BLK) Field Descriptions 2376
17-28. Command Argument Register (SD_ARG) Field Descriptions 2377
17-29. Command and Transfer Mode Register (SD_CMD) Field Descriptions 2378
17-30. Command Response[31:0] Register (SD_RSP10) Field Descriptions 2381
17-31. Command Response[63:32] Register (SD_RSP32) Field Descriptions 2381
17-32. Command Response[95:64] Register (SD_RSP54) Field Descriptions 2382
17-33. Command Response[127:96] Register (SD_RSP76) Field Descriptions 2382
17-34. Data Register (SD_DATA) Field Descriptions 2383
17-35. Present State Register (SD_PSTATE) Field Descriptions 2384
17-36. Control Register (SD_HCTL) Field Descriptions 2387
17-37. SD System Control Register (SD_SYSCTL) Field Descriptions 2390
17-38. Interrupt Status Register (SD_STAT) Field Descriptions 2392
17-39. Interrupt SD Enable Register (SD_IE) Field Descriptions 2397
17-40. Interrupt Signal Enable Register (SD_ISE) Field Descriptions 2400
17-41. Auto CMD12 Error Status Register (SD_AC12) Field Descriptions 2403
17-42. Capabilities Register (SD_CAPA) Field Descriptions 2404
17-43. Maximum Current Capabilities Register (SD_CUR_CAPA) Field Descriptions 2406
17-44. Force Event Register (SD_FE) Field Descriptions 2407
17-45. ADMA Error Status Register (SD_ADMAES) Field Descriptions 2409
17-46. ADMA System Address Low Bits (SD_ADMASAL) Field Descriptions 2410

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17-47. ADMA System Address High Bits Register (SD_ADMASAH) Field Descriptions 2410
17-48. Versions Register (SD_REV) Field Descriptions 2411
18-1. Unsupported UART Features 2414
18-2. UART0 Connectivity Attributes 2416
18-3. UART1–5 Connectivity Attributes 2416
18-4. UART0 Clock Signals 2416
18-5. UART1–5 Clock Signals 2417
18-6. UART Mode Baud and Error Rates 2417
18-7. IrDA Mode Baud and Error Rates 2417
18-8. UART Pin List 2418
18-9. UART Muxing Control 2418
18-10. Local Power-Management Features 2422
18-11. UART Mode Interrupts 2422
18-12. IrDA Mode Interrupts 2423
18-13. CIR Mode Interrupts 2424
18-14. TX FIFO Trigger Level Setting Summary 2427
18-15. RX FIFO Trigger Level Setting Summary 2427
18-16. UART/IrDA/CIR Register Access Mode Programming (Using UART_LCR) 2434
18-17. Subconfiguration Mode A Summary 2435
18-18. Subconfiguration Mode B Summary 2435
18-19. Suboperational Mode Summary 2435
18-20. UART/IrDA/CIR Register Access Mode Overview 2435
18-21. UART Mode Selection 2436
18-22. UART Mode Register Overview 2437
18-23. IrDA Mode Register Overview 2438
18-24. CIR Mode Register Overview 2439
18-25. UART Baud Rate Settings (48-MHz Clock) 2441
18-26. UART Parity Bit Encoding 2442
18-27. UART_EFR[3:0] Software Flow Control Options 2443
18-28. IrDA Baud Rate Settings 2447
18-29. Duty Cycle 2452
18-30. UART Registers 2462
18-31. Receiver Holding Register (RHR) Field Descriptions 2463
18-32. Transmit Holding Register (THR) Field Descriptions 2463
18-33. UART Interrupt Enable Register (IER) Field Descriptions 2464
18-34. IrDA Interrupt Enable Register (IER) Field Descriptions 2465
18-35. CIR Interrupt Enable Register (IER) Field Descriptions 2466
18-36. UART Interrupt Identification Register (IIR) Field Descriptions 2467
18-37. IrDA Interrupt Identification Register (IIR) Field Descriptions 2468
18-38. CIR Interrupt Identification Register (IIR) Field Descriptions 2469
18-39. FIFO Control Register (FCR) Field Descriptions 2470
18-40. Line Control Register (LCR) Field Descriptions 2471
18-41. Modem Control Register (MCR) Field Descriptions 2472
18-42. UART Line Status Register (LSR) Field Descriptions 2473
18-43. IrDA Line Status Register (LSR) Field Descriptions 2474
18-44. CIR Line Status Register (LSR) Field Descriptions 2475
18-45. Modem Status Register (MSR) Field Descriptions 2476
18-46. Transmission Control Register (TCR) Field Descriptions 2477
18-47. Scratchpad Register (SPR) Field Descriptions 2477

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18-48. Trigger Level Register (TLR) Field Descriptions 2478
18-49. RX FIFO Trigger Level Setting Summary 2478
18-50. TX FIFO Trigger Level Setting Summary 2478
18-51. Mode Definition Register 1 (MDR1) Field Descriptions 2479
18-52. Mode Definition Register 2 (MDR2) Field Descriptions 2480
18-53. Status FIFO Line Status Register (SFLSR) Field Descriptions 2481
18-54. RESUME Register Field Descriptions 2481
18-55. Status FIFO Register Low (SFREGL) Field Descriptions 2482
18-56. Status FIFO Register High (SFREGH) Field Descriptions 2482
18-57. BOF Control Register (BLR) Field Descriptions 2483
18-58. Auxiliary Control Register (ACREG) Field Descriptions 2484
18-59. Supplementary Control Register (SCR) Field Descriptions 2485
18-60. Supplementary Status Register (SSR) Field Descriptions 2486
18-61. BOF Length Register (EBLR) Field Descriptions 2487
18-62. Module Version Register (MVR) Field Descriptions 2488
18-63. System Configuration Register (SYSC) Field Descriptions 2489
18-64. System Status Register (SYSS) Field Descriptions 2489
18-65. Wake-Up Enable Register (WER) Field Descriptions 2490
18-66. Carrier Frequency Prescaler Register (CFPS) Field Descriptions 2491
18-67. Divisor Latches Low Register (DLL) Field Descriptions 2492
18-68. Divisor Latches High Register (DLH) Field Descriptions 2492
18-69. Enhanced Feature Register (EFR) Field Descriptions 2493
18-70. EFR[3:0] Software Flow Control Options 2494
18-71. XON1/ADDR1 Register Field Descriptions 2494
18-72. XON2/ADDR2 Register Field Descriptions 2494
18-73. XOFF1 Register Field Descriptions 2495
18-74. XOFF2 Register Field Descriptions 2495
18-75. Transmit Frame Length Low Register (TXFLL) Field Descriptions 2496
18-76. Transmit Frame Length High Register (TXFLH) Field Descriptions 2496
18-77. Received Frame Length Low Register (RXFLL) Field Descriptions 2497
18-78. Received Frame Length High Register (RXFLH) Field Descriptions 2497
18-79. UART Autobauding Status Register (UASR) Field Descriptions 2498
19-1. Timer Resolution and Maximum Range 2500
19-2. Timer[2–7] Connectivity Attributes 2502
19-3. Timer Clock Signals 2502
19-4. Timer Pin List 2503
19-5. Prescaler Functionality 2506
19-6. Prescaler Clock Ratios Value 2509
19-7. Value and Corresponding Interrupt Period 2509
19-8. OCP Error Reporting 2510
19-9. Timer Registers 2513
19-10. Identification Register (TIDR) Field Descriptions 2514
19-11. Timer OCP Configuration Register (TIOCP_CFG) Field Descriptions 2515
19-12. Timer IRQ EOI Register (IRQ_EOI) Field Descriptions 2516
19-13. Timer IRQSTATUS Raw Register (IRQSTATUS_RAW) Field Descriptions 2517
19-14. Timer IRQSTATUS Register (IRQSTATUS) Field Descriptions 2518
19-15. Timer IRQENABLE Set Register (IRQENABLE_SET) Field Descriptions 2519
19-16. Timer IRQENABLE Clear Register (IRQENABLE_CLR) Field Descriptions 2520
19-17. Timer IRQ Wakeup Enable Register (IRQWAKEEN) Field Descriptions 2521

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19-18. Timer Control Register (TCLR) Field Descriptions 2522
19-19. Timer Counter Register (TCRR) Field Descriptions 2523
19-20. Timer Load Register (TLDR) Field Descriptions 2523
19-21. Timer Trigger Register (TTGR) Field Descriptions 2524
19-22. Timer Write Posted Status Register (TWPS) Field Descriptions 2524
19-23. Timer Match Register (TMAR) Field Descriptions 2525
19-24. Timer Capture Register (TCAR1) Field Descriptions 2525
19-25. Timer Synchronous Interface Control Register (TSICR) Field Descriptions 2526
19-26. Timer Capture Register (TCAR2) Field Descriptions 2526
19-27. Timer1 Connectivity Attributes 2528
19-28. Timer Clock Signals 2528
19-29. Value Loaded in TCRR to Generate 1 ms Tick 2529
19-30. Prescaler/Timer Reload Values Versus Contexts 2531
19-31. SmartIdle - Clock Activity Field Configuration 2533
19-32. Prescaler Clock Ratios Value 2533
19-33. Value and Corresponding Interrupt Period 2534
19-34. DMTIMER_1MS REGISTERS 2534
19-35. TIDR Register Field Descriptions 2536
19-36. TIOCP_CFG Register Field Descriptions 2537
19-37. TISTAT Register Field Descriptions 2538
19-38. TISR Register Field Descriptions 2539
19-39. TIER Register Field Descriptions 2540
19-40. TWER Register Field Descriptions 2541
19-41. TCLR Register Field Descriptions 2542
19-42. TCRR Register Field Descriptions 2543
19-43. TLDR Register Field Descriptions 2544
19-44. TTGR Register Field Descriptions 2545
19-45. TWPS Register Field Descriptions 2546
19-46. TMAR Register Field Descriptions 2547
19-47. TCAR1 Register Field Descriptions 2548
19-48. TSICR Register Field Descriptions 2549
19-49. TCAR2 Register Field Descriptions 2550
19-50. TPIR Register Field Descriptions 2551
19-51. TNIR Register Field Descriptions 2552
19-52. TCVR Register Field Descriptions 2553
19-53. TOCR Register Field Descriptions 2554
19-54. TOWR Register Field Descriptions 2555
19-55. RTC Module Connectivity Attributes 2557
19-56. RTC Clock Signals 2557
19-57. RTC Pin List 2557
19-58. RTC Signals 2560
19-59. Interrupt Trigger Events 2561
19-60. RTC Register Names and Values 2563
19-61. Real-Time Clock (RTC) Registers 2567
19-62. Seconds Register (SECONDS_REG) Field Descriptions 2568
19-63. Minutes Register (MINUTES_REG) Field Descriptions 2568
19-64. Hours Register (HOURS_REG) Field Descriptions 2569
19-65. Day of the Month (DAYS_REG) Field Descriptions 2569
19-66. Month Register (MONTHS_REG) Field Descriptions 2570

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19-67. Year Register (YEARS_REG) Field Descriptions 2570
19-68. Day of the Week (WEEKS_REG) Field Descriptions 2571
19-69. Alarm Second Register (ALARM_SECONDS_REG) Field Descriptions 2571
19-70. Alarm Minute Register (ALARM_MINUTES_REG) Field Descriptions 2572
19-71. Alarm Hour Register (ALARM_HOURS_REG) Field Descriptions 2572
19-72. Alarm Day of the Month Register (ALARM_DAYS_REG) Field Descriptions 2573
19-73. Alarm Month Register (ALARM_MONTHS_REG) Field Descriptions 2573
19-74. Alarm Year Register (ALARM_YEARS_REG) Field Descriptions 2574
19-75. Control Register (CTRL_REG) Field Descriptions 2575
19-76. Status Register (STATUS_REG) Field Descriptions 2577
19-77. Interrupt Register (INTERRUPTS_REG) Field Descriptions 2578
19-78. Compensation (LSB) Register (COMP_LSB_REG) Field Descriptions 2579
19-79. Compensation (MSB) Register (COMP_MSB_REG) Field Descriptions 2580
19-80. Oscillator Register (OSC_REG) Field Descriptions 2581
19-81. Scratch Registers (SCRATCHx_REG) Field Descriptions 2581
19-82. Kick0 Register (KICK0R) Field Descriptions 2582
19-83. Kick1 Register (KICK1R) Field Descriptions 2582
19-84. RTC Revision Register (RTC_REVISION) Field Descriptions 2583
19-85. System Configuration Register (RTC_SYSCONFIG) Field Descriptions 2583
19-86. Wakeup Enable Register (RTC_IRQWAKEEN) Field Descriptions 2584
19-87. Public WD Timer Module Connectivity Attributes 2586
19-88. Public WD Timer Clock Signals 2586
19-89. Watchdog Timer Events 2587
19-90. Count and Prescaler Default Reset Values 2588
19-91. Prescaler Clock Ratio Values 2589
19-92. Reset Period Examples 2589
19-93. Default Watchdog Timer Reset Periods 2590
19-94. Global Initialization of Surrounding Modules 2593
19-95. Watchdog Timer Module Global Initialization 2593
19-96. Watchdog Timer Basic Configuration 2593
19-97. Disable the Watchdog Timer 2594
19-98. Enable the Watchdog Timer 2594
19-99. Watchdog Timer Registers 2595
19-100. WDT_WIDR Register Field Descriptions 2596
19-101. WDT_WDSC Register Field Descriptions 2596
19-102. WDT_WDST Register Field Descriptions 2596
19-103. WDT_WISR Register Field Descriptions 2597
19-104. WDT_WIER Register Field Descriptions 2598
19-105. WDT_WWER Field Descriptions 2598
19-106. WDT_WCLR Register Field Descriptions 2599
19-107. WDT_WCRR Register Field Descriptions 2600
19-108. WDT_WLDR Register Field Descriptions 2600
19-109. WDT_WTGR Register Field Descriptions 2600
19-110. WDT_WWPS Register Field Descriptions 2601
19-111. WDT_WDLY Register Field Descriptions 2602
19-112. WDT_WSPR Register Field Descriptions 2602
19-113. WDT_WIRQSTATRAW Register Field Descriptions 2603
19-114. WDT_WIRQSTAT RegisterField Descriptions 2604
19-115. WDT_WIRQENSET Register Field Descriptions 2605

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19-116. WDT_WIRQENCLR Register Field Descriptions 2606
19-117. WDT_WIRQWAKEEN Field Descriptions 2607
20-1. Unsupported I2C Features 2610
20-2. I2C0 Connectivity Attributes 2611
20-3. I2C(1–2) Connectivity Attributes 2611
20-4. I2C Clock Signals 2611
20-5. I2C Pin List 2612
20-6. Signal Pads 2614
20-7. Reset State of I2C Signals 2614
20-8. I2C Registers 2626
20-9. Module Revision Register (LOW BYTES) (I2C_REVNB_LO) Field Descriptions 2627
20-10. Module Revision Register (HIGH BYTES) (I2C_REVNB_HI) Field Descriptions 2628
20-11. System Configuration Register (I2C_SYSC) Field Descriptions 2629
20-12. I2C End of Interrupt Register (I2C_EOI) Field Descriptions 2630
20-13. I2C Status Raw Register (I2C_IRQSTATUS_RAW) Field Descriptions 2630
20-14. I2C Status Register (I2C_IRQSTATUS) Field Descriptions 2635
20-15. I2C Interrupt Enable Set Register (I2C_IRQENABLE_SET) Field Descriptions 2637
20-16. I2C Interrupt Enable Clear Register (I2C_IRQENABLE_CLR) Field Descriptions 2639
20-17. I2C Wakeup Enable Register (I2C_WE) Field Descriptions 2641
20-18. Receive DMA Enable Set Register (I2C_DMARXENABLE_SET) Field Descriptions 2644
20-19. Transmit DMA Enable Set Register (I2C_DMATXENABLE_SET) Field Descriptions 2644
20-20. Receive DMA Enable Clear Register (I2C_DMARXENABLE_CLR) Field Descriptions 2645
20-21. Transmit DMA Enable Clear Register (I2C_DMATXENABLE_CLR) Field Descriptions 2645
20-22. Receive DMA Wakeup Register (I2C_DMARXWAKE_EN) Field Descriptions 2646
20-23. Transmit DMA Wakeup Register (I2C_DMATXWAKE_EN) Field Descriptions 2648
20-24. System Status Register (I2C_SYSS) Field Descriptions 2650
20-25. Buffer Configuration Register (I2C_BUF) Field Descriptions 2650
20-26. Data Counter Register (I2C_CNT) Field Descriptions 2652
20-27. Data Access Register (I2C_DATA) Field Descriptions 2653
20-28. I2C Configuration Register (I2C_CON) Field Descriptions 2654
20-29. I2C Own Address Register (I2C_OA) Field Descriptions 2656
20-30. I2C Slave Address Register (I2C_SA) Field Descriptions 2657
20-31. I2C Clock Prescaler Register (I2C_PSC) Field Descriptions 2658
20-32. I2C SCL Low Time Register (I2C_SCLL) Field Descriptions 2659
20-33. I2C SCL High Time Register (I2C_SCLH) Field Descriptions 2659
20-34. System Test Register (I2C_SYSTEST) Field Descriptions 2660
20-35. I2C Buffer Status Register (I2C_BUFSTAT) Field Descriptions 2663
20-36. Own Address 1 (OA1) (I2C_OA1) Field Descriptions 2664
20-37. I2C Own Address 2 Register (I2C_OA2) Field Descriptions 2665
20-38. I2C Own Address 3 Register (I2C_OA3) Field Descriptions 2666
20-39. Active Own Address Register (I2C_ACTOA) Field Descriptions 2667
20-40. I2C Clock Blocking Enable Register (I2C_SBLOCK) Field Descriptions 2668
21-1. McASP Connectivity Attributes 2672
21-2. McASP Clock Signals 2673
21-3. McASP Pin List 2673
21-4. McASP Interface Signals 2678
21-5. Channel Status and User Data for Each DIT Block 2685
21-6. Transmit Bitstream Data Alignment 2696
21-7. Receive Bitstream Data Alignment 2698

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21-8. McASP Registers Accessed Through Configuration Bus 2717
21-9. McASP Registers Accessed Through Data Port 2719
21-10. McASP AFIFO Registers Accessed Through Peripheral Configuration Port 2720
21-11. Revision Identification Register (REV) Field Descriptions 2720
21-12. Pin Function Register (PFUNC) Field Descriptions 2722
21-13. Pin Direction Register (PDIR) Field Descriptions 2724
21-14. Pin Data Output Register (PDOUT) Field Descriptions 2726
21-15. Pin Data Input Register (PDIN) Field Descriptions 2728
21-16. Pin Data Set Register (PDSET) Field Descriptions 2730
21-17. Pin Data Clear Register (PDCLR) Field Descriptions 2732
21-18. Global Control Register (GBLCTL) Field Descriptions 2733
21-19. Audio Mute Control Register (AMUTE) Field Descriptions 2735
21-20. Digital Loopback Control Register (DLBCTL) Field Descriptions 2737
21-21. Digital Mode Control Register (DITCTL) Field Descriptions 2738
21-22. Receiver Global Control Register (RGBLCTL) Field Descriptions 2739
21-23. Receive Format Unit Bit Mask Register (RMASK) Field Descriptions 2740
21-24. Receive Bit Stream Format Register (RFMT) Field Descriptions 2741
21-25. Receive Frame Sync Control Register (AFSRCTL) Field Descriptions 2743
21-26. Receive Clock Control Register (ACLKRCTL) Field Descriptions 2744
21-27. Receive High-Frequency Clock Control Register (AHCLKRCTL) Field Descriptions 2745
21-28. Receive TDM Time Slot Register (RTDM) Field Descriptions 2746
21-29. Receiver Interrupt Control Register (RINTCTL) Field Descriptions 2747
21-30. Receiver Status Register (RSTAT) Field Descriptions 2748
21-31. Current Receive TDM Time Slot Registers (RSLOT) Field Descriptions 2749
21-32. Receive Clock Check Control Register (RCLKCHK) Field Descriptions 2750
21-33. Receiver DMA Event Control Register (REVTCTL) Field Descriptions 2751
21-34. Transmitter Global Control Register (XGBLCTL) Field Descriptions 2752
21-35. Transmit Format Unit Bit Mask Register (XMASK) Field Descriptions 2753
21-36. Transmit Bit Stream Format Register (XFMT) Field Descriptions 2754
21-37. Transmit Frame Sync Control Register (AFSXCTL) Field Descriptions 2756
21-38. Transmit Clock Control Register (ACLKXCTL) Field Descriptions 2757
21-39. Transmit High-Frequency Clock Control Register (AHCLKXCTL) Field Descriptions 2758
21-40. Transmit TDM Time Slot Register (XTDM) Field Descriptions 2759
21-41. Transmitter Interrupt Control Register (XINTCTL) Field Descriptions 2760
21-42. Transmitter Status Register (XSTAT) Field Descriptions 2761
21-43. Current Transmit TDM Time Slot Register (XSLOT) Field Descriptions 2762
21-44. Transmit Clock Check Control Register (XCLKCHK) Field Descriptions 2763
21-45. Transmitter DMA Event Control Register (XEVTCTL) Field Descriptions 2764
21-46. Serializer Control Registers (SRCTL*n*) Field Descriptions 2765
21-47. Write FIFO Control Register (WFIFOCTL) Field Descriptions 2768
21-48. Write FIFO Status Register (WFIFOSTS) Field Descriptions 2769
21-49. Read FIFO Control Register (RFIFOCTL) Field Descriptions 2770
21-50. Read FIFO Status Register (RFIFOSTS) Field Descriptions 2771
22-1. DCAN Connectivity Attributes 2775
22-2. DCAN Clock Signals 2775
22-3. DCAN Pin List 2775
22-4. Initialization of a Transmit Object 2791
22-5. Initialization of a single Receive Object for Data Frames 2791
22-6. Initialization of a Single Receive Object for Remote Frames 2792

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22-7. Parameters of the CAN Bit Time 2799
22-8. Structure of a Message Object 2809
22-9. Field Descriptions 2809
22-10. Message RAM addressing in Debug/Suspend and RDA Mode 2811
22-11. Message RAM Representation in Debug/Suspend Mode 2812
22-12. Message RAM Representation in RAM Direct Access Mode 2812
22-13. DCAN Control Register Summary Table 2814
22-14. CAN Control Register (DCAN CTL) Field Descriptions 2815
22-15. Error and Status Register (DCAN ES) Field Descriptions 2817
22-16. Parity Error End of Interrupt Register (PARITYERR_EOI) Field Descriptions 2819
22-17. Error Counter Register (DCAN ERRC) Field Descriptions 2819
22-18. Bit Timing Register (DCAN BTR) Field Descriptions 2820
22-19. Interrupt Register (DCAN INT) Field Descriptions 2821
22-20. Test Register (DCAN TEST) Field Descriptions 2822
22-21. Parity Error Code Register (DCAN PERR) Field Descriptions 2823
22-22. Auto-Bus-On Time Register (DCAN ABOTR) Field Descriptions 2824
22-23. Transmission Request Registers (DCAN TXRQ12 to DCAN TXRQ78) 2826
22-24. Transmission Request Registers (DCAN TXRQ12 to DCAN TXRQ78) Field Descriptions 2826
22-25. New Data Registers (DCAN NWDAT12 to DCAN NWDAT78) 2828
22-26. New Data Registers (DCAN NWDAT12 to DCAN NWDAT78) Field Descriptions 2828
22-27. New Data Registers (DCAN NWDAT12 to DCAN NWDAT78) 2830
22-28. New Data Registers (DCAN NWDAT12 to DCAN NWDAT78) Field Descriptions 2830
22-29. Message Valid Registers (DCAN MSGVAL12 to DCAN MSGVAL78) 2832
22-30. Message Valid Registers (DCAN MSGVAL12 to DCAN MSGVAL78) Field Descriptions 2832
22-31. Interrupt Multiplexer Registers (DCAN INTMUX12 to DCAN INTMUX78) 2833
22-32. Interrupt Multiplexer Registers (DCAN INTMUX12 to DCAN INTMUX78) Field Descriptions 2833
22-33. IF1/IF2 Command Registers (DCAN IF1CMD, DCAN IF2CMD) Field Descriptions 2835
22-34. IF1/IF2 Mask Registers (DCAN IF1MSK, DCAN IF2MSK) Field Descriptions 2837
22-35. IF1/IF2 Arbitration Registers (DCAN IF1ARB, DCAN IF2ARB) Field Descriptions 2838
22-36. IF1/IF2 Message Control Registers (DCAN IF1MCTL, DCAN IF2MCTL) Field Descriptions 2840
22-37. IF3 Observation Register (DCAN IF3OBS) Field Descriptions 2843
22-38. IF3 Mask Register (DCAN IF3MSK) Field Descriptions 2845
22-39. IF3 Arbitration Register (DCAN IF3ARB) Field Descriptions 2846
22-40. IF3 Message Control Register (DCAN IF3MCTL) Field Descriptions 2847
22-41. Update Enable Registers (DCAN IF3UPD12 to IF3UPD78)) 2850
22-42. Update Enable Registers (DCAN IF3UPD12 to IF3UPD78) Field Descriptions 2850
22-43. CAN TX I/O Control Register (DCAN TIOC) Field Descriptions 2851
22-44. CAN RX IO Control Register (DCAN RIOC) Field Descriptions 2853
23-1. Unsupported McSPI Features 2856
23-2. McSPI Connectivity Attributes 2858
23-3. McSPI Clock Signals 2858
23-4. McSPI Pin List 2858
23-5. McSPI Interface Pins 2859
23-6. Phase and Polarity Combinations 2863
23-7. Chip Select ↔ Clock Edge Delay Depending on Configuration 2873
23-8. CLKSPIO High/Low Time Computation 2875
23-9. Clock Granularity Examples 2875
23-10. FIFO Writes, Word Length Relationship 2876
23-11. SPI Registers 2895

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23-12. McSPI System Configuration Register (MCSPI_SYSCONFIG) Field Descriptions 2896
23-13. McSPI System Status Register (MCSPI_SYSSTATUS) Field Descriptions 2897
23-14. McSPI Interrupt Status Register (MCSPI_IRQSTATUS) Field Descriptions 2898
23-15. McSPI Interrupt Enable Register (MCSPI_IRQENABLE) Field Descriptions 2901
23-16. McSPI System Register (MCSPI_SYST) Field Descriptions 2903
23-17. McSPI Module Control Register(MCSPI_MODULCTRL) Field Descriptions 2905
23-18. McSPI Channel *(i)* Configuration Register (MCSPI_CH(i)CONF) Field Descriptions 2907
23-19. Data Lines Configurations 2910
23-20. McSPI Channel *(i)* Status Register (MCSPI_CH(i)STAT) Field Descriptions 2911
23-21. McSPI Channel *(i)* Control Register (MCSPI_CH(I)CTRL) Field Descriptions 2912
23-22. McSPI Channel *(i)* Transmit Register (MCSPI_TX*(i)*) Field Descriptions 2913
23-23. McSPI Channel *(i)* Receive Register (MCSPI_RX*(i)*) Field Descriptions 2913
23-24. McSPI Transfer Levels Register (MCSPI_XFERLEVEL) Field Descriptions 2914
24-1. GPIO0 Connectivity Attributes 2918
24-2. GPIO[1:3] Connectivity Attributes 2918
24-3. GPIO Clock Signals 2918
24-4. GPIO Pin List 2919
24-5. GPIO Registers 2930
24-6. GPIO_REVISION Register Field Descriptions 2931
24-7. GPIO_SYSCONFIG Register Field Descriptions 2932
24-8. GPIO_EOI Register Field Descriptions 2933
24-9. GPIO_IRQSTATUS_RAW_n Register Field Descriptions 2934
24-10. GPIO_IRQSTATUS_n Register Field Descriptions 2934
24-11. GPIO_IRQSTATUS_SET_n Register Field Descriptions 2935
24-12. GPIO_IRQSTATUS_CLR_n Register Field Descriptions 2935
24-13. GPIO_IRQWAKEN_n Register Field Descriptions 2936
24-14. GPIO_SYSSTATUS Register Field Descriptions 2937
24-15. GPIO_CTRL Register Field Descriptions 2938
24-16. GPIO_OE Register Field Descriptions 2938
24-17. GPIO_DATAIN Register Field Descriptions 2939
24-18. GPIO_DATAOUT Register Field Descriptions 2939
24-19. GPIO_LEVELDETECT0 Register Field Descriptions 2940
24-20. GPIO_LEVELDETECT1 Register Field Descriptions 2940
24-21. GPIO_RISINGDETECT Register Field Descriptions 2941
24-22. GPIO_FALLINGDETECT Register Field Descriptions 2941
24-23. GPIO_DEBOUNCENABLE Register Field Descriptions 2942
24-24. GPIO_DEBOUNCINGTIME Register Field Descriptions 2942
24-25. GPIO_CLEARDATAOUT Register Field Descriptions 2943
24-26. GPIO_SETDATAOUT Register Field Descriptions 2943
25-1. ROM Exception Vectors 2947
25-2. Dead Loops 2948
25-3. RAM Exception Vectors 2949
25-4. Tracing Data 2949
25-5. Crystal Frequences Supported 2950
25-6. 2950
25-7. 2951
25-8. XIP Timings Parameters 2955
25-9. Pins Used for NOR Boot 2955
25-10. Special SYSBOOT Pins for NOR Boot 2956

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

25-11. NAND Timings Parameters 2957
25-12. ONFI Parameters Page Description 2957
25-13. Supported NAND Devices 2958
25-14. 4th NAND ID Data Byte 2959
25-15. Pins Used for NANDI2C Boot for I2C EEPROM Access 2959
25-16. NAND Geometry Information on I2C EEPROM 2959
25-17. ECC Configuration for NAND Boot 2960
25-18. Pins Used for NAND Boot 2961
25-19. Special SYSBOOT Pins for NAND Boot 2961
25-20. Master Boot Record Structure 2964
25-21. Partition Entry 2964
25-22. Partition Types 2964
25-23. FAT Boot Sector 2965
25-24. FAT Directory Entry 2967
25-25. FAT Entry Description 2968
25-26. Pins Used for MMC0 Boot 2969
25-27. Pins Used for MMC1 Boot 2969
25-28. Pins Used for SPI Boot 2970
25-29. Blocks and Sectors Searched on Non-XIP Memories 2970
25-30. Pins Used for EMAC Boot in MII Mode 2972
25-31. Pins Used for EMAC Boot in RGMII Mode 2972
25-32. Pins Used for EMAC Boot in RMII Mode 2972
25-33. Ethernet PHY Mode Selection 2973
25-34. Pins Used for UART Boot 2973
25-35. Customized Descriptor Parameters 2974
25-36. Pins Used for USB Boot 2974
25-37. GP Device Image Format 2975
25-38. TOC Item Fields 2976
25-39. Filenames in TOC 2977
25-40. Booting Parameters Structure 2978
25-41. Tracing Vectors 2979

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Read This First

About This Manual

This TRM has been designated *Preliminary* because the documentation is in the formative or design phase of development. Texas Instruments reserves the right to change this TRM without notice.

Revision History will not be provided for this TRM as long as it is in the *Preliminary* phase of development.

Related Documentation From Texas Instruments

For a complete listing of related documentation and development-support tools, visit www.ti.com.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Chapter 1

SPRUH73–July 2011

Introduction

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Chapter 2

SPRUH73–July 2011

Interrupts

This section describes the interrupts for the device.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.1 ARM Cortex A8 Interrupts

Table 2-1. ARM Cortex A8 Interrupts

Int Number	Acronym/name	Source	Signal Name
0	EMUINT	MPU Subsystem Internal	Emulation interrupt (EMUICINTR)
1	COMMTX	MPU Subsystem Internal	CortexA8 COMMTX
2	COMMRX	MPU Subsystem Internal	CortexA8 COMMRX
3	BENCH	MPU Subsystem Internal	CortexA8 NPMUIRQ
4	ELM_IRQ	ELM	Sinterrupt (Error location process completion)
5	SSM_WFI_IRQ	MPU Subsystem Internal (SSM)	MPU s/s Secure State Machine Wait for Interrupt (WFI) tracking mechanism
6	SSM_IRQ	MPU Subsystem Internal (SSM)	MPU s/s Secure State Machine interrupt generation for Public FIQ support
7	NMI	External Pin (Primary Input)	nmi_int
8	SEC_EVNT	Firewalls	security_events_irq
9	L3DEBUG	L3	l3_FlagMux_top_FlagOut1
10	L3APPINT	L3	l3_FlagMux_top_FlagOut0
11	PRCMINT	PRCM	irq_mpu
12	EDMACOMPINT	TPCC	tpcc_int_pend_po0
13	EDMAMPERR	TPCC	tpcc_mpint_pend_po
14	EDMAERRINT	TPCC	tpcc_errint_pend_po
15	WDT0INT	WDTIMER0	PO_INT_PEND
16	ADC_TSC_GENINT	ADC_TSC	gen_intr_pend
17	USBSSINT	USBSS	usbss_intr_pend
18	USBINT0	USBSS	usb0_intr_pend
19	USBINT1	USBSS	usb1_intr_pend
20	PRUSS1_EVTOUT0	pr1_host[0] output/events exported from PRUSS1	pr1_host_intr0_intr_pend
21	PRUSS1_EVTOUT1	pr1_host[1] output/events exported from PRUSS1	pr1_host_intr1_intr_pend
22	PRUSS1_EVTOUT2	pr1_host[2] output/events exported from PRUSS1	pr1_host_intr2_intr_pend
23	PRUSS1_EVTOUT3	pr1_host[3] output/events exported from PRUSS1	pr1_host_intr3_intr_pend
24	PRUSS1_EVTOUT4	pr1_host[4] output/events exported from PRUSS1	pr1_host_intr4_intr_pend
25	PRUSS1_EVTOUT5	pr1_host[5] output/events exported from PRUSS1	pr1_host_intr5_intr_pend
26	PRUSS1_EVTOUT6	pr1_host[6] output/events exported from PRUSS1	pr1_host_intr6_intr_pend
27	PRUSS1_EVTOUT7	pr1_host[7] output/events exported from PRUSS1	pr1_host_intr7_intr_pend
28	MMCSD1INT	MMCSD1	SINTERRUPTN
29	MMCSD2INT	MMCSD2	SINTERRUPTN
30	I2C2INT	I2C2	POINTRPEND
31	eCAP0INT	eCAP0 event/interrupt	ecap_intr_intr_pend
32	GPIOINT2A	GPIO 2	POINTRPEND1
33	GPIOINT2B	GPIO 2	POINTRPEND2
34	USBWAKEUP	USBSS	slv0p_Swakeup
36	LCDCINT	LCDC	lcd_irq

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 2-1. ARM Cortex A8 Interrupts (continued)

Int Number	Acronym/name	Source	Signal Name
37	GFXINT	SGX530	THALIAIRQ
38	Reserved		
39	ePWM2INT	eHRPWM2 event/interrupt	epwm_intr_intr_pend
40	3PGSWRXTHR0	3PGSW (GEMAC)	c0_rx_thresh_pend
41	3PGSWRXINT0	3PGSW (GEMAC)	c0_rx_pend
42	3PGSWTXINT0	3PGSW (GEMAC)	c0_tx_pend
43	3PGSWMISC0	3PGSW (GEMAC)	c0_misc_pend
44	UART3INT	UART3	niq
45	UART4INT	UART4	niq
46	UART5INT	UART5	niq
47	eCAP1INT	eCAP1 event/interrupt	ecap_intr_intr_pend
52	DCAN0_INT0	DCAN0	dcan_intr0_intr_pend
53	DCAN0_INT1	DCAN0	dcan_intr1_intr_pend
54	DCAN1_PARITY	DCAN0	dcan_uerr_intr_pend
55	DCAN1_INT0	DCAN1	dcan_intr0_intr_pend
56	DCAN1_INT1	DCAN1	dcan_intr1_intr_pend
57	DCAN1_PARITY	DCAN1	dcan_uerr_intr_pend
58	ePWM0_TZINT	eHRPWM0 TZ interrupt	epwm_tz_intr_pend
59	ePWM1_TZINT	eHRPWM1 TZ interrupt	epwm_tz_intr_pend
60	ePWM2_TZINT	eHRPWM2 TZ interrupt	epwm_tz_intr_pend
61	eCAP2INT	eCAP2 event/interrupt	ecap_intr_intr_pend
62	GPIOINT3A	GPIO 3	POINTRPEND1
63	GPIOINT3B	GPIO 3	POINTRPEND2
64	MMCSD0INT	MMCSD0	SINTERRUPTN
65	SPI0INT	McSPI0	SINTERRUPTN
66	TINT0	Timer0	POINTR_PEND
67	TINT1_1MS	Timer1_MS	POINTR_PEND
68	TINT2	Timer2	POINTR_PEND
69	TINT3	Timer3	POINTR_PEND
70	I2C0INT	I2C0	POINTRPEND
71	I2C1INT	I2C1	POINTRPEND
72	UAR0TINT	UART0	niq
73	UART1INT	UART1	niq
74	UART2INT	UART2	niq
75	RTCINT	RTC	timer_intr_pend
76	RTCALARMINT	RTC	alarm_intr_pend
77	MBINT0	Mailbox0 (mail_u0_irq)	initiator_sinterrupt_q_n0
78	M3_TXEV	Wake M3 Subsystem	TXEV
79	eQEP0INT	eQEP0 event/interrupt	eqep_intr_intr_pend
80	MCATXINT0	McASP0	mcasp_x_intr_pend
81	MCARXINT0	McASP0	mcasp_r_intr_pend
82	MCATXINT1	McASP1	mcasp_x_intr_pend
83	MCARXINT1	McASP1	mcasp_r_intr_pend
84	Reserved		
85	Reserved		
86	ePWM0INT	eHRPWM0 event/interrupt	epwm_intr_intr_pend
87	ePWM1INT	eHRPWM1 event/interrupt	epwm_intr_intr_pend

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 2-1. ARM Cortex A8 Interrupts (continued)

Int Number	Acronym/name	Source	Signal Name
88	eQEP1INT	eQEP1 event/interrupt	eqep_intr_intr_pend
89	eQEP2INT	eQEP2 event/interrupt	eqep_intr_intr_pend
90	DMA_INTR_PIN2	External DMA/Interrupt Pin2	pi_x_dma_event_intr2
91	WDT1INT	WDTIMER1	PO_INT_PEND
92	TINT4	Timer4	POINTR_PEND
93	TINT5	Timer5	POINTR_PEND
94	TINT6	Timer6	POINTR_PEND
95	TINT7	Timer7	POINTR_PEND
96	GPIOINT0A	GPIO 0	POINTRPEND1
97	GPIOINT0B	GPIO 0	POINTRPEND2
98	GPIOINT1A	GPIO 1	POINTRPEND1
99	GPIOINT1B	GPIO 1	POINTRPEND2
100	GPMCINT	GPMC	gpmc_sinterrupt
101	DDRERR0	DDR EMIF0	sys_err_intr_pend
102	AES0_IRQ_S	AES module 0 Interrupt secure side	AES_SINTREQUEST_S
103	AES0_IRQ_P	AES module 0 Interrupt public side	AES_SINTREQUEST_P
104	Reserved		
105	Reserved		
106	Reserved		
107	Reserved		
108	SHA_IRQ_S	SHA2 crypto-accelerator secure side	SHA_SINTREQUEST_S
109	SHA_IRQ_P	SHA2 crypto-accelerator public side	SHA_SINTREQUEST_P
110	FPKA_SINTREQUEST_S	PKA	PKA_SINTREQUEST_S
111	RNG_IRQ	RNG	TRNG_intr_pend
112	TCERRINT0	TPTC0	tptc_erint_pend_po
113	TCERRINT1	TPTC1	tptc_erint_pend_po
114	TCERRINT2	TPTC2	tptc_erint_pend_po
115	ADC_TSC_PENINT	ADC_TSC	pen_intr_pend
116	Reserved		
117	Reserved		
118	Reserved		
119	Reserved		
120	SMRFLX_MPU subsystem	SR0	intrpend
121	SMRFLX_Core	SR1	intrpend
122	Reserved		
123	DMA_INTR_PIN0	External DMA/Interrupt Pin0	pi_x_dma_event_intr0
124	DMA_INTR_PIN1	External DMA/Interrupt Pin1	pi_x_dma_event_intr1
125	SPI1INT	McSPI1	SINTERRUPTN
126	Reserved		
127	Reserved		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.2 ARM Cortex M3 Interrupts

Table 2-2. ARM Cortex M3 Wakeup Processor Interrupts

Int Number	Acronym/Name	Source	Signal Name
0	Reset	M3	
1	NMI	M3	
2	Hard Fault	M3	
3	Mem Manager Fault	M3	
4	Bus Fault	M3	
5	Usage Fault	M3	
6	Reserved		
7	Reserved		
8	Reserved		
9	Reserved		
10	SVCall	M3	
11	Dbug Mon	M3	
12	Reserved		
13	PendSV	M3	
14	SYSTICK	M3	
15	Reserved		
16	PRCM_M3_IRQ1	PRCM	irq_wkup_m3
17	UART0INT	UART0 (L4-WKUP)	nirq
18	TINT0	TIMER0 (L4_WKUP)	POINTR_PEND
19	TINT1_1MS	TIMER1 (L4_WKUP)	POINTR_PEND
20	WDT0INT	WDTimer0 (L4_WKUP)	PO_INT_PEND
21	WDT1INT	WDTimer1 (L4_WKUP)	PO_INT_PEND
22	I2CI0NT	I2C0 (L4_WKUP)	POINTRPEND
23	GPIOINT0A	GPIO0 (L4_WKUP)	POINTRPEND1
24	GPIOINT0A	GPIO0 (L4_WKUP)	POINTRPEND2
25			
26	ADC_TSC_INT1	ADC_TSC (L4_WKUP)	gen_intr_pend
27	RTCINT	RTC	timer_intr_pend
28	RTCALARMINT	RTC	alarm_intr_pend
29	SMRFLX_MPU subsystem	SR0	intrpend
30	SMRFLX_Core	SR1	intrpend
31	MBINT0	Mailbox0 (mail_u3_irq)	initiator_sinterrupt_q_n3
32			
33	USBSWAKEUP	USBSS	slv0p_Swakeup
34	PRCM_M3_IRQ2	PRCM	irq_mpuisgated_wkup_m3
35	USB0WUOUT	USBSS	usb0_wuout
36	USB1WUOUT	USBSS	usb1_wuout
37	DMA_INTR_PIN1	External DMA/Interrupt Pin1	pi_x_dma_event_intr1
38	DMA_INTR_PIN2	External DMA/Interrupt Pin2	pi_x_dma_event_intr2
39			
40	I2C0_WAKE	I2C0 (L4_WKUP)	POINTRSWAKEUP
41	RTCTIMER_WAKE	RTC Timer Wake	slvp_timer_SWakeup
42	RTCALARM_WAKE	RTC Alarm Wake	slvp_alarm_SWakeup
43	TIMER0_WAKE	TIMER0 (L4_WKUP)	POINTR_SWAKEUP
44	TIMER1_WAKE	TIMER1 (L4_WKUP)	PO_SWAKEUP_OUT

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 2-2. ARM Cortex M3 Wakeup Processor Interrupts (continued)

Int Number	Acronym/Name	Source	Signal Name
45	UART0_WAKE	UART0 (L4-WKUP)	swakeup
46	GPIO0_WAKE0	GPIO0 (L4_WKUP)	POINTRSWAKEUP1
47	GPIO0_WAKE1	GPIO0 (L4_WKUP)	POINTRSWAKEUP2
48	MPU_WAKE	MPU subsystem	POMWAKEUP
49	WDT0_WAKE	WDTimer0 (L4_WKUP)	PO_INT_WAKEUP
50	WDT1_WAKE	WDTimer1 (L4_WKUP)	PO_INT_WAKEUP
51	ADCTSC_WAKE	ADC_TSC (L4_WKUP)	pwr_idle_prm_Swakeup

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.3 PRUSS Interrupts

Table 2-3. PRUSS1 Interrupts

Int Number	Signal Name (Non Ethercat Mode)	Source	Signal Name (Ethercat Mode)
63	tpcc_int_pend_po1	TPCC	
62	tpcc_errint_pend_po	TPCC	
61	tptc_erint_pend_po	TPTC0	
60	initiator_sinterrupt_q_n1	Mbox0 - mail_u1_irq (mailbox interrupt for pru0)	
59	initiator_sinterrupt_q_n2	Mbox0 - mail_u2_irq (mailbox interrupt for pru1)	
58	Emulation Suspend Signal (software use)	Debugss	
57	POINTRPEND1	GPIO0	
56	pwm_trip_zone	eHRPWM0/eHRPWM1/eHRPWM2	
55	mcasp_x_intr_pend	McASP0 Tx	pr1_mii1_crs(external)
54	mcasp_r_intr_pend	McASP0 Rx	PRU1_RX_EOF
53	gen_intr_pend	ADC_TSC	MDIO_MII_LINK[1]
52	nirq	UART2	PORT1_TX_OVERFLOW
51	nirq	UART0	PORT1_TX_UNDERFLOW
50	c0_rx_thresh_pend	3PGSW (GEMAC)	PRU1_RX_OVERFLOW
49	c0_rx_pend	3PGSW (GEMAC)	PRU1_RX_NIBBLE_ODD
48	c0_tx_pend	3PGSW (GEMAC)	PRU1_RX_CRC
47	c0_misc_pend	3PGSW (GEMAC)	PRU1_RX_SOF
46	epwm_intr_intr_pend	eHRPWM1	PRU1_RX_SFD
45	eqep_intr_intr_pend	eQEP0	PRU1_RX_ERR32
44	SINTERRUPTN	McSPI0	PRU1_RX_ERR
43	epwm_intr_intr_pend	eHRPWM0	pr1_mii0_crs(external)
42	ecap_intr_intr_pend	eCAP0	PRU0_RX_EOF
41	POINTRPEND	I2C0	MDIO_MII_LINK[0]
40	dcan_intr	DCAN0	PORT0_TX_OVERFLOW
39	dcan_int1	DCAN0	PORT0_TX_UNDERFLOW
38	dcan_uerr	DCAN0	PRU0_RX_OVERFLOW
37	epwm_intr_intr_pend	eHRPWM2	PRU0_RX_NIBBLE_ODD
36	ecap_intr_intr_pend	eCAP2	PRU0_RX_CRC
35	ecap_intr_intr_pend	eCAP1	PRU0_RX_SOF
34	mcasp_r_intr_pend	McASP1 Rx	PRU0_RX_SFD
33	mcasp_x_intr_pend	McASP1 Tx	PRU0_RX_ERR32
32	nirq	UART1	PRU0_RX_ERR
31			
30			
29			
28			
27			
26			
25			
24			
23	pr1_pru_mst_intr<15:0>_intr_req	pru0 or pru1	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 2-3. PRUSS1 Interrupts (continued)

Int Number	Signal Name (Non Ethercat Mode)	Source	Signal Name (Ethercat Mode)
22			
21			
20			
19			
18			
17			
16			
15	pr1_ecap_intr_req	PRUSS eCAP	
14	sync0_out_pend	PRUSS Ethercat	
13	sync1_out_pend	PRUSS Ethercat	
12	latch0_in(input to PRUSS)	PRUSS Ethercat	
11	latch1_in(input to PRUSS)	PRUSS Ethercat	
10	pdi_wd_exp_pend	PRUSS Ethercat	
9	pd_wd_exp_pend	PRUSS Ethercat	
8	digio_event_req	PRUSS Ethercat	
7	pr1_iep_tim_cap_cmp_pend		
6	pr1_uart6_uint_intr_req		
5	pr1_uart5_utxevt_intr_req		
4	pr1_uart4_urxevt_intr_req		
3	pr1_xfr_timeout		
2	pr1_pru1_r31_status_cnt16		
1	pr1_pru0_r31_status_cnt16		
0	pr1_parity_err_intr_pend	PRUSS Parity Logic	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.4 EDMA Events

Table 2-4. EDMA Events

Event Count	Event Name	Source Module
0	SDTXEVT0	MMCHS0
1	SDRXEVT0	MMCHS0
2	SDTXEVT1	MMCHS1
3	SDRXEVT1	MMCHS1
4	SDTXEVT1	MMCHS2
5	SDRXEVT1	MMCHS2
6	I2CTXEVT0	I2C0
7	I2CRXEVT0	I2C0
8	I2CTXEVT1	I2C1
9	I2CRXEVT1	I2C1
10	I2CTXEVT2	I2C2
11	I2CRXEVT2	I2C2
12	UTXEVT0	UART0
13	URXEVT0	UART0
14	UTXEVT1	UART1
15	URXEVT1	UART1
16	UTXEVT2	UART2
17	URXEVT2	UART2
18	UTXEVT3	UART3
19	URXEVT3	UART3
20	UTXEVT4	UART4
21	URXEVT4	UART4
22	UTXEVT5	UART5
23	URXEVT5	UART5
24		
25		
26	CAN_IF1DMA	DCAN 0
27	CAN_IF2DMA	DCAN 0
28	CAN_IF3DMA	DCAN 0
29	CAN_IF1DMA	DCAN 1
30	CAN_IF2DMA	DCAN 1
31	CAN_IF3DMA	DCAN 1
32	SPIXEVT0	McSPI0
33	SPIREVT0	McSPI0
34	SPIXEVT1	McSPI0
35	SPIREVT1	McSPI0
36	SPIXEVT0	McSPI1
37	SPIREVT0	McSPI1
38	SPIXEVT1	McSPI1
39	SPIREVT1	McSPI1
40	Reserved	Reserved
41	Reserved	Reserved
42	Reserved	Reserved
43	Reserved	Reserved
44	Reserved	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 2-4. EDMA Events (continued)

Event Count	Event Name	Source Module
45	Reserved	Reserved
46	Reserved	Reserved
47	Reserved	Reserved
48	TINT0	Timer 0
49	TINT1	Timer 1
50	TINT2	Timer 2
51	TINT3	Timer 3
52	TINT4	Timer 4
53	TINT5	Timer 5
54	TINT6	Timer 6
55	TINT7	Timer 7
56	GPMCEVT	GPMC
57	AXEVT0	McASP0
58	AREVT0	McASP0
59	AXEVT1	McASP1
60	AREVT1	McASP1
61	Reserved	Reserved
62	Reserved	Reserved
63	eCAPEVT0	eCAP 0
64	eCAPEVT1	eCAP 1
65	eCAPEVT2	eCAP 2
66	eHRPWMEVT0	eHRPWM 0
67	eHRPWMEVT1	eHRPWM 1
68	eHRPWMEVT2	eHRPWM 2
69	eQEPEVT0	eQEP 0
70	eQEPEVT1	eQEP 1
71	eQEPEVT2	eQEP 2
72	pr1_host[7]	PRUSS1
73	pr1_host[6]	PURSS1
74	tsc_adc_FIFOevent_dpend	ADC/TSC
75	tsc_adc_FIFO1event_dpend	ADC/TSC
76	AES0_p_dma_ctx_in_req	AES Module 0: Request a new context on the public HIB
77	AES0_p_dma_data_in_req	AES Module 0: Request input data on the public HIB
78	AES0_p_dma_data_out_req	AES Module 0: Request output data read on the public HIB
79	Reserved	Reserved
80	Reserved	Reserved
81	Reserved	Reserved
82	Reserved	Reserved
83	Reserved	Reserved
84	Reserved	Reserved
85	SHA2_dma_ctxin_p	SHA2MD5 Module 1: Request a context on public HIB
86	SHA2_dma_din_p	SHA2MD5 Module 1: Request input data on public HIB
87	SHA2_dma_ctxout_p	SHA2MD5 Module 1: Request output data/context on public HIB

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 2-4. EDMA Events (continued)

Event Count	Event Name	Source Module
88	AES0_p_dma_context_out_req	AES Module 0: Request the authentication result (TAG) or result IV read on public HIB
89	Reserved	Reserved
90	pi_x_dma_event_intr0	external pin
91	pi_x_dma_event_intr1	external pin
92	pi_x_dma_event_intr2	external pin
93	GPIOEVT0	GPIO BANK 0
94	GPIOEVT1	GPIO BANK 1
95	GPIOEVT2	GPIO BANK 2

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.5 Crypto DMA Events

Table 2-5. Crypto DMA Events

Event#	Event Name	Source Module
1	AES0_s_dma_ctx_in_req	AES Module 0: Request a new context on the Secure HIB
2	AES0_s_dma_data_in_req	AES Module 0: Request input data on the Secured HIB
3	AES0_s_dma_data_out_req	AES Module 0: Request output data read on the Secured HIB
4	AES0_p_dma_ctx_in_req	AES Module 0: Request a new context on the Public HIB
5	AES0_p_dma_data_in_req	AES Module 0: Request input data on the Public HIB
6	AES0_p_dma_data_out_req	AES Module 0: Request output data read on the Public HIB
7	AES1_s_dma_ctx_in_req	AES Module 1: Request a new context on the Secured HIB
8	AES1_s_dma_data_in_req	AES Module 1: Request input data on the Secured HIB
9	AES1_s_dma_data_out_req	AES Module 1: Request output data read on the Secured HIB
10	AES1_p_dma_ctx_in_req	AES Module 1: Request a new context on the Public HIB
11	AES1_p_dma_data_in_req	AES Module 1: Request input data on the Public HIB
12	AES1_p_dma_data_out_req	AES Module 1: Request output data read on the Public HIB
13	Reserved	
14	Reserved	
15	DES_s_dma_ctx_in_req	DES Module: Request a new context on the secure HIB
16	DES_s_dma_data_in_req	DES Module: Request input data on the secure HIB
17	DES_s_dma_data_out_req	DES Module: Request output data read on the secure HIB
18	DES_p_dma_ctx_in_req	DES Module: Request a new context on the public HIB
19	DES_p_dma_data_in_req	DES Module: Request input data on the public HIB
20	DES_p_dma_data_out_req	DES Module: Request output data read on the public HIB
21	SHA2_dma_ctxin_s	SHA2MD5 Module 1: Request a context on secure HIB
22	SHA2_dma_din_s	SHA2MD5 Module 1: Request input data on secure HIB
23	SHA2_dma_ctxout_s	SHA2MD5 Module 1: Request output data/context on secure HIB
24	SHA2_dma_ctxin_p	SHA2MD5 Module 1: Request a context on public HIB
25	SHA2_dma_din_p	SHA2MD5 Module 1: Request input data on public HIB
26	SHA2_dma_ctxout_p	SHA2MD5 Module 1: Request output data/context on public HIB
27	AES0_s_dma_context_out_req	AES Module 0: Request the authentication result (TAG) or result IV read on the Secured HIB

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 2-5. Crypto DMA Events (continued)

Event#	Event Name	Source Module
28	AES0_p_dma_context_out_req	AES Module 0: Request the authentication result (TAG) or result IV read on the Public HIB
29	AES1_s_dma_context_out_req	AES Module 1: Request the authentication result (TAG) or result IV read on the Secured HIB
30	AES1_p_dma_context_out_req	AES Module 1: Request the authentication result (TAG) or result IV read on the Public HIB
31	Reserved	
32	Reserved	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.6 PWM Events

Table 2-6. Timer & eCAP Event Capture

Event #	IP	Interrupt Name/Pin
0	For Timer 5 MUX input from IO signal TIMER5	TIMER5 IO pin
	For Timer 6 MUX input from IO signal TIMER6	TIMER6 IO pin
	For Timer 7 MUX input from IO signal TIMER7	TIMER7 IO pin
	For eCAP 0 MUX input from IO signal eCAP0	eCAP0 IO pin
	For eCAP 1 MUX input from IO signal eCAP1	eCAP1 IO pin
	For eCAP 2 MUX input from IO signal eCAP2	eCAP2 IO pin
1	UART0	UART0INT
2	UART1	UART1INT
3	UART2	UART2INT
4	UART3	UART3INT
5	UART4	UART4INT
6	UART5	UART5INT
7	3PGSW	3PGSWRXTHR0
8	3PGSW	3PGSWRXINT0
9	3PGSW	3PGSWTXINT0
10	3PGSW	3PGSWMISC0
11	McASP0	MCATXINT0
12	McASP0	MCARXINT0
13	McASP1	MCATXINT1
14	McASP1	MCARXINT1
15	McASP2	MCATXINT2
16	McASP2	MCARXINT2
17	GPIO 0	GPIOINT0A
18	GPIO 0	GPIOINT0B
19	GPIO 1	GPIOINT1A
20	GPIO 1	GPIOINT1B
21	GPIO 2	GPIOINT2A
22	GPIO 2	GPIOINT2B
23	GPIO 3	GPIOINT3A
24	GPIO 3	GPIOINT3B
25	DCAN0	DCAN0_INT0
26	DCAN0	DCAN0_INT1
27	DCAN0	DCAN0_PARITY
28	DCAN1	DCAN1_INT0
29	DCAN1	DCAN1_INT1
30	DCAN1	DCAN1_PARITY

2.7 INTCPS Registers

Table 2-7 lists the memory-mapped registers for the INTCPS. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 2-7 should be considered as reserved locations and the register contents should not be modified.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 2-7. INTCPS REGISTERS

Offset	Acronym	Register Description	Section
0h	INTCPS_REVISION	This register contains the IP revision code	Section 2.7.1
64h	INTCPS_FIQ_PRIORITY	This register supplies the currently active FIQ priority level	Section 2.7.2
68h	INTCPS_THRESHOLD	This register sets the priority threshold	Section 2.7.3
80h	INTCPS_ITR0	This register shows the raw interrupt input status before masking	Section 2.7.4
84h	INTCPS_MIR0	This register contains the interrupt mask	Section 2.7.5
88h	INTCPS_MIR_CLEAR0	This register is used to clear the interrupt mask bits.	Section 2.7.6
8Ch	INTCPS_MIR_SET0	This register is used to set the interrupt mask bits.	Section 2.7.7
90h	INTCPS_ISR_SET0	This register is used to set the software interrupt bits. It is also used to read the currently active software interrupts.	Section 2.7.8
94h	INTCPS_ISR_CLEAR0	This register is used to clear the software interrupt bits	Section 2.7.9
98h	INTCPS_PENDING_IRQ0	This register contains the IRQ status after masking	Section 2.7.10
9Ch	INTCPS_PENDING_FIQ0	This register contains the FIQ status after masking	Section 2.7.11
10h	INTCPS_SYSCONFIG	This register controls the various parameters of the OCP interface	Section 2.7.12
A0h	INTCPS_ITR1	This register shows the raw interrupt input status before masking	Section 2.7.13
A4h	INTCPS_MIR1	This register contains the interrupt mask	Section 2.7.14
A8h	INTCPS_MIR_CLEAR1	This register is used to clear the interrupt mask bits.	Section 2.7.15
ACh	INTCPS_MIR_SET1	This register is used to set the interrupt mask bits.	Section 2.7.16
B0h	INTCPS_ISR_SET1	This register is used to set the software interrupt bits. It is also used to read the currently active software interrupts.	Section 2.7.17
B4h	INTCPS_ISR_CLEAR1	This register is used to clear the software interrupt bits	Section 2.7.18
B8h	INTCPS_PENDING_IRQ1	This register contains the IRQ status after masking	Section 2.7.19
BCh	INTCPS_PENDING_FIQ1	This register contains the FIQ status after masking	Section 2.7.20
C0h	INTCPS_ITR2	This register shows the raw interrupt input status before masking	Section 2.7.21
C4h	INTCPS_MIR2	This register contains the interrupt mask	Section 2.7.22
14h	INTCPS_SYSSTATUS	This register provides status information about the module	Section 2.7.23
C8h	INTCPS_MIR_CLEAR2	This register is used to clear the interrupt mask bits.	Section 2.7.24
CCh	INTCPS_MIR_SET2	This register is used to set the interrupt mask bits.	Section 2.7.25
D0h	INTCPS_ISR_SET2	This register is used to set the software interrupt bits. It is also used to read the currently active software interrupts.	Section 2.7.26
D4h	INTCPS_ISR_CLEAR2	This register is used to clear the software interrupt bits	Section 2.7.27
D8h	INTCPS_PENDING_IRQ2	This register contains the IRQ status after masking	Section 2.7.28
DCh	INTCPS_PENDING_FIQ2	This register contains the FIQ status after masking	Section 2.7.29
E0h	INTCPS_ITR3	This register shows the raw interrupt input status before masking	Section 2.7.30
E4h	INTCPS_MIR3	This register contains the interrupt mask	Section 2.7.31
E8h	INTCPS_MIR_CLEAR3	This register is used to clear the interrupt mask bits.	Section 2.7.32
ECh	INTCPS_MIR_SET3	This register is used to set the interrupt mask bits.	Section 2.7.33
F0h	INTCPS_ISR_SET3	This register is used to set the software interrupt bits. It is also used to read the currently active software interrupts.	Section 2.7.34
F4h	INTCPS_ISR_CLEAR3	This register is used to clear the software interrupt bits	Section 2.7.35
F8h	INTCPS_PENDING_IRQ3	This register contains the IRQ status after masking	Section 2.7.36
FCh	INTCPS_PENDING_FIQ3	This register contains the FIQ status after masking	Section 2.7.37

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 2-7. INTCPS REGISTERS (continued)

Offset	Acronym	Register Description	Section
100h	INTCPS_ILR0	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.38
104h	INTCPS_ILR1	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.39
108h	INTCPS_ILR2	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.40
10Ch	INTCPS_ILR3	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.41
110h	INTCPS_ILR4	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.42
114h	INTCPS_ILR5	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.43
118h	INTCPS_ILR6	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.44
11Ch	INTCPS_ILR7	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.45
120h	INTCPS_ILR8	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.46
124h	INTCPS_ILR9	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.47
128h	INTCPS_ILR10	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.48
12Ch	INTCPS_ILR11	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.49
130h	INTCPS_ILR12	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.50
134h	INTCPS_ILR13	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.51
138h	INTCPS_ILR14	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.52
13Ch	INTCPS_ILR15	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.53
140h	INTCPS_ILR16	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.54
144h	INTCPS_ILR17	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.55
148h	INTCPS_ILR18	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.56
14Ch	INTCPS_ILR19	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.57
150h	INTCPS_ILR20	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.58
154h	INTCPS_ILR21	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.59
158h	INTCPS_ILR22	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.60
15Ch	INTCPS_ILR23	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.61
160h	INTCPS_ILR24	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.62
164h	INTCPS_ILR25	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.63
168h	INTCPS_ILR26	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.64
16Ch	INTCPS_ILR27	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.65

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 2-7. INTCPS REGISTERS (continued)

Offset	Acronym	Register Description	Section
170h	INTCPS_ILR28	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.66
174h	INTCPS_ILR29	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.67
178h	INTCPS_ILR30	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.68
17Ch	INTCPS_ILR31	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.69
180h	INTCPS_ILR32	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.70
184h	INTCPS_ILR33	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.71
188h	INTCPS_ILR34	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.72
18Ch	INTCPS_ILR35	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.73
190h	INTCPS_ILR36	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.74
194h	INTCPS_ILR37	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.75
198h	INTCPS_ILR38	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.76
19Ch	INTCPS_ILR39	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.77
1A0h	INTCPS_ILR40	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.78
1A4h	INTCPS_ILR41	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.79
1A8h	INTCPS_ILR42	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.80
1ACh	INTCPS_ILR43	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.81
1B0h	INTCPS_ILR44	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.82
1B4h	INTCPS_ILR45	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.83
1B8h	INTCPS_ILR46	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.84
1BCh	INTCPS_ILR47	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.85
1C0h	INTCPS_ILR48	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.86
1C4h	INTCPS_ILR49	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.87
1C8h	INTCPS_ILR50	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.88
1CCh	INTCPS_ILR51	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.89
1D0h	INTCPS_ILR52	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.90
1D4h	INTCPS_ILR53	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.91
1D8h	INTCPS_ILR54	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.92
1DCh	INTCPS_ILR55	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.93

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 2-7. INTCPS REGISTERS (continued)

Offset	Acronym	Register Description	Section
1E0h	INTCPS_ILR56	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.94
1E4h	INTCPS_ILR57	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.95
1E8h	INTCPS_ILR58	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.96
1ECh	INTCPS_ILR59	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.97
1F0h	INTCPS_ILR60	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.98
1F4h	INTCPS_ILR61	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.99
1F8h	INTCPS_ILR62	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.100
1FCh	INTCPS_ILR63	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.101
200h	INTCPS_ILR64	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.102
204h	INTCPS_ILR65	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.103
208h	INTCPS_ILR66	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.104
20Ch	INTCPS_ILR67	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.105
210h	INTCPS_ILR68	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.106
214h	INTCPS_ILR69	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.107
218h	INTCPS_ILR70	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.108
21Ch	INTCPS_ILR71	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.109
220h	INTCPS_ILR72	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.110
224h	INTCPS_ILR73	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.111
228h	INTCPS_ILR74	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.112
22Ch	INTCPS_ILR75	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.113
230h	INTCPS_ILR76	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.114
234h	INTCPS_ILR77	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.115
238h	INTCPS_ILR78	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.116
23Ch	INTCPS_ILR79	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.117
240h	INTCPS_ILR80	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.118
244h	INTCPS_ILR81	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.119
248h	INTCPS_ILR82	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.120
24Ch	INTCPS_ILR83	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.121

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 2-7. INTCPS REGISTERS (continued)

Offset	Acronym	Register Description	Section
250h	INTCPS_ILR84	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.122
254h	INTCPS_ILR85	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.123
258h	INTCPS_ILR86	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.124
25Ch	INTCPS_ILR87	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.125
260h	INTCPS_ILR88	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.126
264h	INTCPS_ILR89	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.127
268h	INTCPS_ILR90	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.128
26Ch	INTCPS_ILR91	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.129
270h	INTCPS_ILR92	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.130
274h	INTCPS_ILR93	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.131
278h	INTCPS_ILR94	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.132
27Ch	INTCPS_ILR95	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.133
40h	INTCPS_SIR_IRQ	This register supplies the currently active IRQ interrupt number.	Section 2.7.134
280h	INTCPS_ILR96	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.135
284h	INTCPS_ILR97	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.136
288h	INTCPS_ILR98	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.137
28Ch	INTCPS_ILR99	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.138
290h	INTCPS_ILR100	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.139
294h	INTCPS_ILR101	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.140
298h	INTCPS_ILR102	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.141
29Ch	INTCPS_ILR103	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.142
2A0h	INTCPS_ILR104	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.143
2A4h	INTCPS_ILR105	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.144
44h	INTCPS_SIR_FIQ	This register supplies the currently active FIQ interrupt number	Section 2.7.145
2A8h	INTCPS_ILR106	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.146
2ACh	INTCPS_ILR107	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.147
2B0h	INTCPS_ILR108	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.148
2B4h	INTCPS_ILR109	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.149

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 2-7. INTCPS REGISTERS (continued)

Offset	Acronym	Register Description	Section
2B8h	INTCPS_ILR110	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.150
2BCh	INTCPS_ILR111	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.151
2C0h	INTCPS_ILR112	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.152
2C4h	INTCPS_ILR113	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.153
2C8h	INTCPS_ILR114	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.154
2CCh	INTCPS_ILR115	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.155
48h	INTCPS_CONTROL	This register contains the new interrupt agreement bits	Section 2.7.156
2D0h	INTCPS_ILR116	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.157
2D4h	INTCPS_ILR117	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.158
2D8h	INTCPS_ILR118	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.159
2DCh	INTCPS_ILR119	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.160
2E0h	INTCPS_ILR120	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.161
2E4h	INTCPS_ILR121	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.162
2E8h	INTCPS_ILR122	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.163
2ECh	INTCPS_ILR123	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.164
2F0h	INTCPS_ILR124	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.165
2F4h	INTCPS_ILR125	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.166
4Ch	INTCPS_PROTECTION	This register controls protection of the other registers. This register can only be accessed in priviledged mode, regardless of the curent value of the protection bit.	Section 2.7.167
2F8h	INTCPS_ILR126	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.168
2FCh	INTCPS_ILR127	These registers contain the priority for the interrupts and the FIQ/IRQ steering.	Section 2.7.169
50h	INTCPS_IDLE	This register controls the clock auto-idle for the functional clock and the input synchronisers	Section 2.7.170
60h	INTCPS_IRQ_PRIORITY	This register supplies the currently active IRQ priority level	Section 2.7.171

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.1 INTCPS_REVISION Register (offset = 0h) [reset = 0h]

This register contains the IP revision code

INTCPS_REVISION is shown in Figure 2-1 and described in Table 2-8.

Figure 2-1. INTCPS_REVISION Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Rev							
R							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-8. INTCPS_REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	Reserved	R		Reads returns 0
7-0	Rev	R		IP revision [7:4] Major revision [3:0] Minor revision Examples: 0x10 for 1.0, 0x21 for 2.1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.2 INTCPS_FIQ_PRIORITY Register (offset = 64h) [reset = 0h]

This register supplies the currently active FIQ priority level

INTCPS_FIQ_PRIORITY is shown in Figure 2-2 and described in Table 2-9.

Figure 2-2. INTCPS_FIQ_PRIORITY Register

31	30	29	28	27	26	25	24
SpuriousFIQflag							
R							

23	22	21	20	19	18	17	16
SpuriousFIQflag							
R							

15	14	13	12	11	10	9	8
SpuriousFIQflag							
R							

7	6	5	4	3	2	1	0
SpuriousFIQflag	FIQPriority						
R	R						

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-9. INTCPS_FIQ_PRIORITY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	SpuriousFIQflag	R		Spurious FIQ flag
6-0	FIQPriority	R		Current FIQ priority

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.3 INTCPS_THRESHOLD Register (offset = 68h) [reset = 0h]

This register sets the priority threshold

INTCPS_THRESHOLD is shown in Figure 2-3 and described in Table 2-10.

Figure 2-3. INTCPS_THRESHOLD Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
PriorityThreshold							
R/W							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-10. INTCPS_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	Reserved	R		Reads returns 0
7-0	PriorityThreshold	R/W		Priority threshold (used values 0x00-0x1f or 0x00-0x3f, 0xff disables the threshold).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.4 INTCPS_ITR0 Register (offset = 80h) [reset = 0h]

This register shows the raw interrupt input status before masking

INTCPS_ITR0 is shown in Figure 2-4 and described in Table 2-11.

Figure 2-4. INTCPS_ITR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Itr																															
R																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-11. INTCPS_ITR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Itr	R		Interrupt status before masking

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.5 INTCPS_MIR0 Register (offset = 84h) [reset = 0h]

This register contains the interrupt mask

INTCPS_MIR0 is shown in Figure 2-5 and described in Table 2-12.

Figure 2-5. INTCPS_MIR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mir																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-12. INTCPS_MIR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Mir	R/W		Interrupt mask

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.6 INTCPS_MIR_CLEAR0 Register (offset = 88h) [reset = 0h]

This register is used to clear the interrupt mask bits.

INTCPS_MIR_CLEAR0 is shown in Figure 2-6 and described in Table 2-13.

Figure 2-6. INTCPS_MIR_CLEAR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MirClear																															
W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 2-13. INTCPS_MIR_CLEAR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MirClear	W		Write 1 clears the mask bit to 0, reads return 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.7 INTCPS_MIR_SET0 Register (offset = 8Ch) [reset = 0h]

This register is used to set the interrupt mask bits.

INTCPS_MIR_SET0 is shown in Figure 2-7 and described in Table 2-14.

Figure 2-7. INTCPS_MIR_SET0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MirSet																															
W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-14. INTCPS_MIR_SET0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MirSet	W		Write 1 sets the mask bit to 1, reads return 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.8 INTCPS_ISR_SET0 Register (offset = 90h) [reset = 0h]

This register is used to set the software interrupt bits. It is also used to read the currently active software interrupts.

INTCPS_ISR_SET0 is shown in Figure 2-8 and described in Table 2-15.

Figure 2-8. INTCPS_ISR_SET0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IsrSet																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-15. INTCPS_ISR_SET0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IsrSet	R/W		Reads returns the currently active software interrupts, Write 1 sets the software interrupt bits to 1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.9 INTCPS_ISR_CLEAR0 Register (offset = 94h) [reset = 0h]

This register is used to clear the software interrupt bits

INTCPS_ISR_CLEAR0 is shown in Figure 2-9 and described in Table 2-16.

Figure 2-9. INTCPS_ISR_CLEAR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IsrClear																															
W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-16. INTCPS_ISR_CLEAR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IsrClear	W		Write 1 clears the sofware interrupt bits to 0, reads return 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.10 INTCPS_PENDING_IRQ0 Register (offset = 98h) [reset = 0h]

This register contains the IRQ status after masking

INTCPS_PENDING_IRQ0 is shown in Figure 2-10 and described in Table 2-17.

Figure 2-10. INTCPS_PENDING_IRQ0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PendingIRQ																															
R																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 2-17. INTCPS_PENDING_IRQ0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PendingIRQ	R		IRQ status after masking

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.11 INTCPS_PENDING_FIQ0 Register (offset = 9Ch) [reset = 0h]

This register contains the FIQ status after masking

INTCPS_PENDING_FIQ0 is shown in Figure 2-11 and described in Table 2-18.

Figure 2-11. INTCPS_PENDING_FIQ0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PendingFIQ																															
R																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-18. INTCPS_PENDING_FIQ0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PendingFIQ	R		FIQ status after masking

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.12 INTCPS_SYSCONFIG Register (offset = 10h) [reset = 0h]

This register controls the various parameters of the OCP interface

INTCPS_SYSCONFIG is shown in Figure 2-12 and described in Table 2-19.

Figure 2-12. INTCPS_SYSCONFIG Register

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			Reserved		Reserved	SoftReset	Autoidle
			R			R/W	R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 2-19. INTCPS_SYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
4-3	Reserved	R		Write 0's for future compatibility. Reads returns 0
1	SoftReset	R/W		Software reset. Set this bit to trigger a module reset. The bit is automatically reset by the hardware. During reads, it always returns 0. 0 - nofun_w: no functional effect 0 - always_r: Always returns 0 1 - rstMode_w: The module is reset 1 - never_r: never happens
0	Autoidle	R/W		Internal OCP clock gating strategy 0 - clkfree: OCP clock is free running 1 - autoClkGate: Automatic OCP clock gating strategy is applied, bnased on the OCP interface activity

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.13 INTCPS_ITR1 Register (offset = A0h) [reset = 0h]

This register shows the raw interrupt input status before masking

INTCPS_ITR1 is shown in Figure 2-13 and described in Table 2-20.

Figure 2-13. INTCPS_ITR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Itr																															
R																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-20. INTCPS_ITR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Itr	R		Interrupt status before masking

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.14 INTCPS_MIR1 Register (offset = A4h) [reset = 0h]

This register contains the interrupt mask

INTCPS_MIR1 is shown in Figure 2-14 and described in Table 2-21.

Figure 2-14. INTCPS_MIR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mir																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-21. INTCPS_MIR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Mir	R/W		Interrupt mask

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.15 INTCPS_MIR_CLEAR1 Register (offset = A8h) [reset = 0h]

This register is used to clear the interrupt mask bits.

INTCPS_MIR_CLEAR1 is shown in Figure 2-15 and described in Table 2-22.

Figure 2-15. INTCPS_MIR_CLEAR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MirClear																															
W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-22. INTCPS_MIR_CLEAR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MirClear	W		Write 1 clears the mask bit to 0, reads return 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.16 INTCPS_MIR_SET1 Register (offset = ACh) [reset = 0h]

This register is used to set the interrupt mask bits.

INTCPS_MIR_SET1 is shown in Figure 2-16 and described in Table 2-23.

Figure 2-16. INTCPS_MIR_SET1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MirSet																															
W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-23. INTCPS_MIR_SET1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MirSet	W		Write 1 sets the mask bit to 1, reads return 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.17 INTCPS_ISR_SET1 Register (offset = B0h) [reset = 0h]

This register is used to set the software interrupt bits. It is also used to read the currently active software interrupts.

INTCPS_ISR_SET1 is shown in Figure 2-17 and described in Table 2-24.

Figure 2-17. INTCPS_ISR_SET1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IsrSet																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-24. INTCPS_ISR_SET1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IsrSet	R/W		Reads returns the currently active software interrupts, Write 1 sets the software interrupt bits to 1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.18 INTCPS_ISR_CLEAR1 Register (offset = B4h) [reset = 0h]

This register is used to clear the software interrupt bits

INTCPS_ISR_CLEAR1 is shown in Figure 2-18 and described in Table 2-25.

Figure 2-18. INTCPS_ISR_CLEAR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IsrClear																															
W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-25. INTCPS_ISR_CLEAR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IsrClear	W		Write 1 clears the sofware interrupt bits to 0, reads return 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.19 INTCPS_PENDING_IRQ1 Register (offset = B8h) [reset = 0h]

This register contains the IRQ status after masking

INTCPS_PENDING_IRQ1 is shown in Figure 2-19 and described in Table 2-26.

Figure 2-19. INTCPS_PENDING_IRQ1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PendingIRQ																															
R																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 2-26. INTCPS_PENDING_IRQ1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PendingIRQ	R		IRQ status after masking

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.20 INTCPS_PENDING_FIQ1 Register (offset = BCh) [reset = 0h]

This register contains the FIQ status after masking

INTCPS_PENDING_FIQ1 is shown in Figure 2-20 and described in Table 2-27.

Figure 2-20. INTCPS_PENDING_FIQ1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PendingFIQ																															
R																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 2-27. INTCPS_PENDING_FIQ1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PendingFIQ	R		FIQ status after masking

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.21 INTCPS_ITR2 Register (offset = C0h) [reset = 0h]

This register shows the raw interrupt input status before masking

INTCPS_ITR2 is shown in Figure 2-21 and described in Table 2-28.

Figure 2-21. INTCPS_ITR2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Itr																															
R																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 2-28. INTCPS_ITR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Itr	R		Interrupt status before masking

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.22 INTCPS_MIR2 Register (offset = C4h) [reset = 0h]

This register contains the interrupt mask

INTCPS_MIR2 is shown in Figure 2-22 and described in Table 2-29.

Figure 2-22. INTCPS_MIR2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mir																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-29. INTCPS_MIR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Mir	R/W		Interrupt mask

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.23 INTCPS_SYSSTATUS Register (offset = 14h) [reset = 0h]

This register provides status information about the module

INTCPS_SYSSTATUS is shown in Figure 2-23 and described in Table 2-30.

Figure 2-23. INTCPS_SYSSTATUS Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved							ResetDone
			R				R

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-30. INTCPS_SYSSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
7-1	Reserved	R		Reserved for OCP socket status information Read returns 0
0	ResetDone	R		Internal reset monitoring 0 - rstOngoing: Internal module reset is on-going 1 - rstComp: Reset completed

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.24 INTCPS_MIR_CLEAR2 Register (offset = C8h) [reset = 0h]

This register is used to clear the interrupt mask bits.

INTCPS_MIR_CLEAR2 is shown in Figure 2-24 and described in Table 2-31.

Figure 2-24. INTCPS_MIR_CLEAR2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MirClear																															
W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-31. INTCPS_MIR_CLEAR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MirClear	W		Write 1 clears the mask bit to 0, reads return 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.25 INTCPS_MIR_SET2 Register (offset = CCh) [reset = 0h]

This register is used to set the interrupt mask bits.

INTCPS_MIR_SET2 is shown in Figure 2-25 and described in Table 2-32.

Figure 2-25. INTCPS_MIR_SET2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MirSet																															
W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-32. INTCPS_MIR_SET2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MirSet	W		Write 1 sets the mask bit to 1, reads return 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.26 INTCPS_ISR_SET2 Register (offset = D0h) [reset = 0h]

This register is used to set the software interrupt bits. It is also used to read the currently active software interrupts.

INTCPS_ISR_SET2 is shown in Figure 2-26 and described in Table 2-33.

Figure 2-26. INTCPS_ISR_SET2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IsrSet																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-33. INTCPS_ISR_SET2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IsrSet	R/W		Reads returns the currently active software interrupts, Write 1 sets the software interrupt bits to 1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.27 INTCPS_ISR_CLEAR2 Register (offset = D4h) [reset = 0h]

This register is used to clear the software interrupt bits

INTCPS_ISR_CLEAR2 is shown in Figure 2-27 and described in Table 2-34.

Figure 2-27. INTCPS_ISR_CLEAR2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IsrClear																															
W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-34. INTCPS_ISR_CLEAR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IsrClear	W		Write 1 clears the sofware interrupt bits to 0, reads return 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.28 INTCPS_PENDING_IRQ2 Register (offset = D8h) [reset = 0h]

This register contains the IRQ status after masking

INTCPS_PENDING_IRQ2 is shown in Figure 2-28 and described in Table 2-35.

Figure 2-28. INTCPS_PENDING_IRQ2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PendingIRQ																															
R																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-35. INTCPS_PENDING_IRQ2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PendingIRQ	R		IRQ status after masking

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.29 INTCPS_PENDING_FIQ2 Register (offset = DCh) [reset = 0h]

This register contains the FIQ status after masking

INTCPS_PENDING_FIQ2 is shown in Figure 2-29 and described in Table 2-36.

Figure 2-29. INTCPS_PENDING_FIQ2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PendingFIQ																															
R																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-36. INTCPS_PENDING_FIQ2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PendingFIQ	R		FIQ status after masking

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.30 INTCPS_ITR3 Register (offset = E0h) [reset = 0h]

This register shows the raw interrupt input status before masking

INTCPS_ITR3 is shown in Figure 2-30 and described in Table 2-37.

Figure 2-30. INTCPS_ITR3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Itr																															
R																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-37. INTCPS_ITR3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Itr	R		Interrupt status before masking

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.31 INTCPS_MIR3 Register (offset = E4h) [reset = 0h]

This register contains the interrupt mask

INTCPS_MIR3 is shown in Figure 2-31 and described in Table 2-38.

Figure 2-31. INTCPS_MIR3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mir																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-38. INTCPS_MIR3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	Mir	R/W		Interrupt mask

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.32 INTCPS_MIR_CLEAR3 Register (offset = E8h) [reset = 0h]

This register is used to clear the interrupt mask bits.

INTCPS_MIR_CLEAR3 is shown in Figure 2-32 and described in Table 2-39.

Figure 2-32. INTCPS_MIR_CLEAR3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MirClear																															
W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 2-39. INTCPS_MIR_CLEAR3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MirClear	W		Write 1 clears the mask bit to 0, reads return 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.33 INTCPS_MIR_SET3 Register (offset = ECh) [reset = 0h]

This register is used to set the interrupt mask bits.

INTCPS_MIR_SET3 is shown in Figure 2-33 and described in Table 2-40.

Figure 2-33. INTCPS_MIR_SET3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MirSet																															
W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-40. INTCPS_MIR_SET3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MirSet	W		Write 1 sets the mask bit to 1, reads return 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.34 INTCPS_ISR_SET3 Register (offset = F0h) [reset = 0h]

This register is used to set the software interrupt bits. It is also used to read the currently active software interrupts.

INTCPS_ISR_SET3 is shown in Figure 2-34 and described in Table 2-41.

Figure 2-34. INTCPS_ISR_SET3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IsrSet																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-41. INTCPS_ISR_SET3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IsrSet	R/W		Reads returns the currently active software interrupts, Write 1 sets the software interrupt bits to 1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.35 INTCPS_ISR_CLEAR3 Register (offset = F4h) [reset = 0h]

This register is used to clear the software interrupt bits

INTCPS_ISR_CLEAR3 is shown in Figure 2-35 and described in Table 2-42.

Figure 2-35. INTCPS_ISR_CLEAR3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IsrClear																															
W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-42. INTCPS_ISR_CLEAR3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	IsrClear	W		Write 1 clears the sofware interrupt bits to 0, reads return 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.36 INTCPS_PENDING_IRQ3 Register (offset = F8h) [reset = 0h]

This register contains the IRQ status after masking

INTCPS_PENDING_IRQ3 is shown in Figure 2-36 and described in Table 2-43.

Figure 2-36. INTCPS_PENDING_IRQ3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PendingIRQ																															
R																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-43. INTCPS_PENDING_IRQ3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PendingIRQ	R		IRQ status after masking

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.37 INTCPS_PENDING_FIQ3 Register (offset = FCh) [reset = 0h]

This register contains the FIQ status after masking

INTCPS_PENDING_FIQ3 is shown in Figure 2-37 and described in Table 2-44.

Figure 2-37. INTCPS_PENDING_FIQ3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PendingFIQ																															
R																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-44. INTCPS_PENDING_FIQ3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PendingFIQ	R		FIQ status after masking

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.38 INTCPS_ILR0 Register (offset = 100h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR0 is shown in Figure 2-38 and described in Table 2-45.

Figure 2-38. INTCPS_ILR0 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-45. INTCPS_ILR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.39 INTCPS_ILR1 Register (offset = 104h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR1 is shown in Figure 2-39 and described in Table 2-46.

Figure 2-39. INTCPS_ILR1 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-46. INTCPS_ILR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.40 INTCPS_ILR2 Register (offset = 108h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR2 is shown in Figure 2-40 and described in Table 2-47.

Figure 2-40. INTCPS_ILR2 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-47. INTCPS_ILR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.41 INTCPS_ILR3 Register (offset = 10Ch) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR3 is shown in Figure 2-41 and described in Table 2-48.

Figure 2-41. INTCPS_ILR3 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-48. INTCPS_ILR3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.42 INTCPS_ILR4 Register (offset = 110h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR4 is shown in Figure 2-42 and described in Table 2-49.

Figure 2-42. INTCPS_ILR4 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-49. INTCPS_ILR4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.43 INTCPS_ILR5 Register (offset = 114h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR5 is shown in Figure 2-43 and described in Table 2-50.

Figure 2-43. INTCPS_ILR5 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-50. INTCPS_ILR5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.44 INTCPS_ILR6 Register (offset = 118h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR6 is shown in Figure 2-44 and described in Table 2-51.

Figure 2-44. INTCPS_ILR6 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-51. INTCPS_ILR6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.45 INTCPS_ILR7 Register (offset = 11Ch) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR7 is shown in Figure 2-45 and described in Table 2-52.

Figure 2-45. INTCPS_ILR7 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-52. INTCPS_ILR7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.46 INTCPS_ILR8 Register (offset = 120h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR8 is shown in Figure 2-46 and described in Table 2-53.

Figure 2-46. INTCPS_ILR8 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-53. INTCPS_ILR8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.47 INTCPS_ILR9 Register (offset = 124h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR9 is shown in Figure 2-47 and described in Table 2-54.

Figure 2-47. INTCPS_ILR9 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-54. INTCPS_ILR9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.48 INTCPS_ILR10 Register (offset = 128h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR10 is shown in Figure 2-48 and described in Table 2-55.

Figure 2-48. INTCPS_ILR10 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-55. INTCPS_ILR10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.49 INTCPS_ILR11 Register (offset = 12Ch) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR11 is shown in Figure 2-49 and described in Table 2-56.

Figure 2-49. INTCPS_ILR11 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-56. INTCPS_ILR11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.50 INTCPS_ILR12 Register (offset = 130h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR12 is shown in Figure 2-50 and described in Table 2-57.

Figure 2-50. INTCPS_ILR12 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-57. INTCPS_ILR12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.51 INTCPS_ILR13 Register (offset = 134h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR13 is shown in Figure 2-51 and described in Table 2-58.

Figure 2-51. INTCPS_ILR13 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-58. INTCPS_ILR13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.52 INTCPS_ILR14 Register (offset = 138h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR14 is shown in Figure 2-52 and described in Table 2-59.

Figure 2-52. INTCPS_ILR14 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-59. INTCPS_ILR14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.53 INTCPS_ILR15 Register (offset = 13Ch) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR15 is shown in Figure 2-53 and described in Table 2-60.

Figure 2-53. INTCPS_ILR15 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-60. INTCPS_ILR15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.54 INTCPS_ILR16 Register (offset = 140h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR16 is shown in Figure 2-54 and described in Table 2-61.

Figure 2-54. INTCPS_ILR16 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-61. INTCPS_ILR16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.55 INTCPS_ILR17 Register (offset = 144h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR17 is shown in Figure 2-55 and described in Table 2-62.

Figure 2-55. INTCPS_ILR17 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-62. INTCPS_ILR17 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.56 INTCPS_ILR18 Register (offset = 148h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR18 is shown in Figure 2-56 and described in Table 2-63.

Figure 2-56. INTCPS_ILR18 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-63. INTCPS_ILR18 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.57 INTCPS_ILR19 Register (offset = 14Ch) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR19 is shown in Figure 2-57 and described in Table 2-64.

Figure 2-57. INTCPS_ILR19 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-64. INTCPS_ILR19 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.58 INTCPS_ILR20 Register (offset = 150h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR20 is shown in Figure 2-58 and described in Table 2-65.

Figure 2-58. INTCPS_ILR20 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-65. INTCPS_ILR20 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.59 INTCPS_ILR21 Register (offset = 154h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR21 is shown in Figure 2-59 and described in Table 2-66.

Figure 2-59. INTCPS_ILR21 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-66. INTCPS_ILR21 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.60 INTCPS_ILR22 Register (offset = 158h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR22 is shown in Figure 2-60 and described in Table 2-67.

Figure 2-60. INTCPS_ILR22 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-67. INTCPS_ILR22 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.61 INTCPS_ILR23 Register (offset = 15Ch) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR23 is shown in Figure 2-61 and described in Table 2-68.

Figure 2-61. INTCPS_ILR23 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-68. INTCPS_ILR23 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.62 INTCPS_ILR24 Register (offset = 160h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR24 is shown in Figure 2-62 and described in Table 2-69.

Figure 2-62. INTCPS_ILR24 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-69. INTCPS_ILR24 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.63 INTCPS_ILR25 Register (offset = 164h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR25 is shown in Figure 2-63 and described in Table 2-70.

Figure 2-63. INTCPS_ILR25 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-70. INTCPS_ILR25 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.64 INTCPS_ILR26 Register (offset = 168h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR26 is shown in Figure 2-64 and described in Table 2-71.

Figure 2-64. INTCPS_ILR26 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-71. INTCPS_ILR26 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.65 INTCPS_ILR27 Register (offset = 16Ch) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR27 is shown in Figure 2-65 and described in Table 2-72.

Figure 2-65. INTCPS_ILR27 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-72. INTCPS_ILR27 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.66 INTCPS_ILR28 Register (offset = 170h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR28 is shown in Figure 2-66 and described in Table 2-73.

Figure 2-66. INTCPS_ILR28 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-73. INTCPS_ILR28 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.67 INTCPS_ILR29 Register (offset = 174h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR29 is shown in Figure 2-67 and described in Table 2-74.

Figure 2-67. INTCPS_ILR29 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-74. INTCPS_ILR29 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.68 INTCPS_ILR30 Register (offset = 178h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR30 is shown in Figure 2-68 and described in Table 2-75.

Figure 2-68. INTCPS_ILR30 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-75. INTCPS_ILR30 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.69 INTCPS_ILR31 Register (offset = 17Ch) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR31 is shown in Figure 2-69 and described in Table 2-76.

Figure 2-69. INTCPS_ILR31 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-76. INTCPS_ILR31 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.70 INTCPS_ILR32 Register (offset = 180h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR32 is shown in Figure 2-70 and described in Table 2-77.

Figure 2-70. INTCPS_ILR32 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-77. INTCPS_ILR32 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.71 INTCPS_ILR33 Register (offset = 184h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR33 is shown in Figure 2-71 and described in Table 2-78.

Figure 2-71. INTCPS_ILR33 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-78. INTCPS_ILR33 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.72 INTCPS_ILR34 Register (offset = 188h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR34 is shown in Figure 2-72 and described in Table 2-79.

Figure 2-72. INTCPS_ILR34 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-79. INTCPS_ILR34 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.73 INTCPS_ILR35 Register (offset = 18Ch) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR35 is shown in Figure 2-73 and described in Table 2-80.

Figure 2-73. INTCPS_ILR35 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-80. INTCPS_ILR35 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.74 INTCPS_ILR36 Register (offset = 190h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR36 is shown in Figure 2-74 and described in Table 2-81.

Figure 2-74. INTCPS_ILR36 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-81. INTCPS_ILR36 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.75 INTCPS_ILR37 Register (offset = 194h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR37 is shown in Figure 2-75 and described in Table 2-82.

Figure 2-75. INTCPS_ILR37 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-82. INTCPS_ILR37 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.76 INTCPS_ILR38 Register (offset = 198h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR38 is shown in Figure 2-76 and described in Table 2-83.

Figure 2-76. INTCPS_ILR38 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-83. INTCPS_ILR38 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.77 INTCPS_ILR39 Register (offset = 19Ch) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR39 is shown in Figure 2-77 and described in Table 2-84.

Figure 2-77. INTCPS_ILR39 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-84. INTCPS_ILR39 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.78 INTCPS_ILR40 Register (offset = 1A0h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR40 is shown in Figure 2-78 and described in Table 2-85.

Figure 2-78. INTCPS_ILR40 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-85. INTCPS_ILR40 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.79 INTCPS_ILR41 Register (offset = 1A4h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR41 is shown in Figure 2-79 and described in Table 2-86.

Figure 2-79. INTCPS_ILR41 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-86. INTCPS_ILR41 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.80 INTCPS_ILR42 Register (offset = 1A8h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR42 is shown in Figure 2-80 and described in Table 2-87.

Figure 2-80. INTCPS_ILR42 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-87. INTCPS_ILR42 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.81 INTCPS_ILR43 Register (offset = 1ACh) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR43 is shown in Figure 2-81 and described in Table 2-88.

Figure 2-81. INTCPS_ILR43 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-88. INTCPS_ILR43 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.82 INTCPS_ILR44 Register (offset = 1B0h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR44 is shown in Figure 2-82 and described in Table 2-89.

Figure 2-82. INTCPS_ILR44 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-89. INTCPS_ILR44 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.83 INTCPS_ILR45 Register (offset = 1B4h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR45 is shown in Figure 2-83 and described in Table 2-90.

Figure 2-83. INTCPS_ILR45 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-90. INTCPS_ILR45 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.84 INTCPS_ILR46 Register (offset = 1B8h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR46 is shown in Figure 2-84 and described in Table 2-91.

Figure 2-84. INTCPS_ILR46 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-91. INTCPS_ILR46 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.85 INTCPS_ILR47 Register (offset = 1BCh) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR47 is shown in Figure 2-85 and described in Table 2-92.

Figure 2-85. INTCPS_ILR47 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-92. INTCPS_ILR47 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.86 INTCPS_ILR48 Register (offset = 1C0h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR48 is shown in Figure 2-86 and described in Table 2-93.

Figure 2-86. INTCPS_ILR48 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-93. INTCPS_ILR48 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.87 INTCPS_ILR49 Register (offset = 1C4h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR49 is shown in Figure 2-87 and described in Table 2-94.

Figure 2-87. INTCPS_ILR49 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-94. INTCPS_ILR49 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.88 INTCPS_ILR50 Register (offset = 1C8h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR50 is shown in Figure 2-88 and described in Table 2-95.

Figure 2-88. INTCPS_ILR50 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-95. INTCPS_ILR50 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.89 INTCPS_ILR51 Register (offset = 1CCh) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR51 is shown in Figure 2-89 and described in Table 2-96.

Figure 2-89. INTCPS_ILR51 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-96. INTCPS_ILR51 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.90 INTCPS_ILR52 Register (offset = 1D0h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR52 is shown in Figure 2-90 and described in Table 2-97.

Figure 2-90. INTCPS_ILR52 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-97. INTCPS_ILR52 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.91 INTCPS_ILR53 Register (offset = 1D4h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR53 is shown in Figure 2-91 and described in Table 2-98.

Figure 2-91. INTCPS_ILR53 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-98. INTCPS_ILR53 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.92 INTCPS_ILR54 Register (offset = 1D8h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR54 is shown in Figure 2-92 and described in Table 2-99.

Figure 2-92. INTCPS_ILR54 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-99. INTCPS_ILR54 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.93 INTCPS_ILR55 Register (offset = 1DCh) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR55 is shown in Figure 2-93 and described in Table 2-100.

Figure 2-93. INTCPS_ILR55 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-100. INTCPS_ILR55 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.94 INTCPS_ILR56 Register (offset = 1E0h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR56 is shown in Figure 2-94 and described in Table 2-101.

Figure 2-94. INTCPS_ILR56 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-101. INTCPS_ILR56 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.95 INTCPS_ILR57 Register (offset = 1E4h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR57 is shown in Figure 2-95 and described in Table 2-102.

Figure 2-95. INTCPS_ILR57 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-102. INTCPS_ILR57 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.96 INTCPS_ILR58 Register (offset = 1E8h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR58 is shown in Figure 2-96 and described in Table 2-103.

Figure 2-96. INTCPS_ILR58 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-103. INTCPS_ILR58 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.97 INTCPS_ILR59 Register (offset = 1ECh) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR59 is shown in Figure 2-97 and described in Table 2-104.

Figure 2-97. INTCPS_ILR59 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-104. INTCPS_ILR59 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.98 INTCPS_ILR60 Register (offset = 1F0h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR60 is shown in Figure 2-98 and described in Table 2-105.

Figure 2-98. INTCPS_ILR60 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-105. INTCPS_ILR60 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.99 INTCPS_ILR61 Register (offset = 1F4h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR61 is shown in Figure 2-99 and described in Table 2-106.

Figure 2-99. INTCPS_ILR61 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-106. INTCPS_ILR61 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.100 INTCPS_ILR62 Register (offset = 1F8h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR62 is shown in Figure 2-100 and described in Table 2-107.

Figure 2-100. INTCPS_ILR62 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-107. INTCPS_ILR62 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.101 INTCPS_ILR63 Register (offset = 1FCh) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR63 is shown in Figure 2-101 and described in Table 2-108.

Figure 2-101. INTCPS_ILR63 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-108. INTCPS_ILR63 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.102 INTCPS_ILR64 Register (offset = 200h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR64 is shown in Figure 2-102 and described in Table 2-109.

Figure 2-102. INTCPS_ILR64 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-109. INTCPS_ILR64 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.103 INTCPS_ILR65 Register (offset = 204h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR65 is shown in Figure 2-103 and described in Table 2-110.

Figure 2-103. INTCPS_ILR65 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-110. INTCPS_ILR65 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.104 INTCPS_ILR66 Register (offset = 208h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR66 is shown in Figure 2-104 and described in Table 2-111.

Figure 2-104. INTCPS_ILR66 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-111. INTCPS_ILR66 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.105 INTCPS_ILR67 Register (offset = 20Ch) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR67 is shown in Figure 2-105 and described in Table 2-112.

Figure 2-105. INTCPS_ILR67 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-112. INTCPS_ILR67 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.106 INTCPS_ILR68 Register (offset = 210h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR68 is shown in Figure 2-106 and described in Table 2-113.

Figure 2-106. INTCPS_ILR68 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-113. INTCPS_ILR68 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.107 INTCPS_ILR69 Register (offset = 214h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR69 is shown in Figure 2-107 and described in Table 2-114.

Figure 2-107. INTCPS_ILR69 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-114. INTCPS_ILR69 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.108 INTCPS_ILR70 Register (offset = 218h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR70 is shown in Figure 2-108 and described in Table 2-115.

Figure 2-108. INTCPS_ILR70 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-115. INTCPS_ILR70 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.109 INTCPS_ILR71 Register (offset = 21Ch) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR71 is shown in Figure 2-109 and described in Table 2-116.

Figure 2-109. INTCPS_ILR71 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-116. INTCPS_ILR71 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.110 INTCPS_ILR72 Register (offset = 220h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR72 is shown in Figure 2-110 and described in Table 2-117.

Figure 2-110. INTCPS_ILR72 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-117. INTCPS_ILR72 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.111 INTCPS_ILR73 Register (offset = 224h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR73 is shown in Figure 2-111 and described in Table 2-118.

Figure 2-111. INTCPS_ILR73 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-118. INTCPS_ILR73 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.112 INTCPS_ILR74 Register (offset = 228h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR74 is shown in Figure 2-112 and described in Table 2-119.

Figure 2-112. INTCPS_ILR74 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-119. INTCPS_ILR74 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.113 INTCPS_ILR75 Register (offset = 22Ch) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR75 is shown in Figure 2-113 and described in Table 2-120.

Figure 2-113. INTCPS_ILR75 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-120. INTCPS_ILR75 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.114 INTCPS_ILR76 Register (offset = 230h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR76 is shown in Figure 2-114 and described in Table 2-121.

Figure 2-114. INTCPS_ILR76 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-121. INTCPS_ILR76 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.115 INTCPS_ILR77 Register (offset = 234h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR77 is shown in Figure 2-115 and described in Table 2-122.

Figure 2-115. INTCPS_ILR77 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-122. INTCPS_ILR77 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.116 INTCPS_ILR78 Register (offset = 238h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR78 is shown in Figure 2-116 and described in Table 2-123.

Figure 2-116. INTCPS_ILR78 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-123. INTCPS_ILR78 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.117 INTCPS_ILR79 Register (offset = 23Ch) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR79 is shown in Figure 2-117 and described in Table 2-124.

Figure 2-117. INTCPS_ILR79 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-124. INTCPS_ILR79 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.118 INTCPS_ILR80 Register (offset = 240h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR80 is shown in Figure 2-118 and described in Table 2-125.

Figure 2-118. INTCPS_ILR80 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-125. INTCPS_ILR80 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.119 INTCPS_ILR81 Register (offset = 244h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR81 is shown in Figure 2-119 and described in Table 2-126.

Figure 2-119. INTCPS_ILR81 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-126. INTCPS_ILR81 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.120 INTCPS_ILR82 Register (offset = 248h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR82 is shown in Figure 2-120 and described in Table 2-127.

Figure 2-120. INTCPS_ILR82 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-127. INTCPS_ILR82 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.121 INTCPS_ILR83 Register (offset = 24Ch) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR83 is shown in Figure 2-121 and described in Table 2-128.

Figure 2-121. INTCPS_ILR83 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-128. INTCPS_ILR83 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.122 INTCPS_ILR84 Register (offset = 250h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR84 is shown in Figure 2-122 and described in Table 2-129.

Figure 2-122. INTCPS_ILR84 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-129. INTCPS_ILR84 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.123 INTCPS_ILR85 Register (offset = 254h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR85 is shown in Figure 2-123 and described in Table 2-130.

Figure 2-123. INTCPS_ILR85 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-130. INTCPS_ILR85 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.124 INTCPS_ILR86 Register (offset = 258h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR86 is shown in Figure 2-124 and described in Table 2-131.

Figure 2-124. INTCPS_ILR86 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-131. INTCPS_ILR86 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.125 INTCPS_ILR87 Register (offset = 25Ch) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR87 is shown in Figure 2-125 and described in Table 2-132.

Figure 2-125. INTCPS_ILR87 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-132. INTCPS_ILR87 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.126 INTCPS_ILR88 Register (offset = 260h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR88 is shown in Figure 2-126 and described in Table 2-133.

Figure 2-126. INTCPS_ILR88 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-133. INTCPS_ILR88 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.127 INTCPS_ILR89 Register (offset = 264h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR89 is shown in Figure 2-127 and described in Table 2-134.

Figure 2-127. INTCPS_ILR89 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-134. INTCPS_ILR89 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.128 INTCPS_ILR90 Register (offset = 268h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR90 is shown in Figure 2-128 and described in Table 2-135.

Figure 2-128. INTCPS_ILR90 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-135. INTCPS_ILR90 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.129 INTCPS_ILR91 Register (offset = 26Ch) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR91 is shown in Figure 2-129 and described in Table 2-136.

Figure 2-129. INTCPS_ILR91 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-136. INTCPS_ILR91 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.130 INTCPS_ILR92 Register (offset = 270h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR92 is shown in Figure 2-130 and described in Table 2-137.

Figure 2-130. INTCPS_ILR92 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-137. INTCPS_ILR92 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.131 INTCPS_ILR93 Register (offset = 274h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR93 is shown in Figure 2-131 and described in Table 2-138.

Figure 2-131. INTCPS_ILR93 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-138. INTCPS_ILR93 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.132 INTCPS_ILR94 Register (offset = 278h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR94 is shown in Figure 2-132 and described in Table 2-139.

Figure 2-132. INTCPS_ILR94 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-139. INTCPS_ILR94 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.133 INTCPS_ILR95 Register (offset = 27Ch) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR95 is shown in Figure 2-133 and described in Table 2-140.

Figure 2-133. INTCPS_ILR95 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-140. INTCPS_ILR95 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.134 INTCPS_SIR_IRQ Register (offset = 40h) [reset = 0h]

This register supplies the currently active IRQ interrupt number.

INTCPS_SIR_IRQ is shown in Figure 2-134 and described in Table 2-141.

Figure 2-134. INTCPS_SIR_IRQ Register

31	30	29	28	27	26	25	24
SpuriousIRQ							
R/W							

23	22	21	20	19	18	17	16
SpuriousIRQ							
R/W							

15	14	13	12	11	10	9	8
SpuriousIRQ							
R/W							

7	6	5	4	3	2	1	0
SpuriousIRQ	ActiveIRQ						
R/W	R/W						

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-141. INTCPS_SIR_IRQ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	SpuriousIRQ	R/W		Spurious IRQ flag
6-0	ActiveIRQ	R/W		Active IRQ number

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.135 INTCPS_ILR96 Register (offset = 280h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR96 is shown in Figure 2-135 and described in Table 2-142.

Figure 2-135. INTCPS_ILR96 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-142. INTCPS_ILR96 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.136 INTCPS_ILR97 Register (offset = 284h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR97 is shown in Figure 2-136 and described in Table 2-143.

Figure 2-136. INTCPS_ILR97 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-143. INTCPS_ILR97 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.137 INTCPS_ILR98 Register (offset = 288h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR98 is shown in Figure 2-137 and described in Table 2-144.

Figure 2-137. INTCPS_ILR98 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-144. INTCPS_ILR98 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.138 INTCPS_ILR99 Register (offset = 28Ch) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR99 is shown in Figure 2-138 and described in Table 2-145.

Figure 2-138. INTCPS_ILR99 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-145. INTCPS_ILR99 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.139 INTCPS_ILR100 Register (offset = 290h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR100 is shown in Figure 2-139 and described in Table 2-146.

Figure 2-139. INTCPS_ILR100 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-146. INTCPS_ILR100 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.140 INTCPS_ILR101 Register (offset = 294h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR101 is shown in Figure 2-140 and described in Table 2-147.

Figure 2-140. INTCPS_ILR101 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-147. INTCPS_ILR101 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.141 INTCPS_ILR102 Register (offset = 298h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR102 is shown in Figure 2-141 and described in Table 2-148.

Figure 2-141. INTCPS_ILR102 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-148. INTCPS_ILR102 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.142 INTCPS_ILR103 Register (offset = 29Ch) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR103 is shown in Figure 2-142 and described in Table 2-149.

Figure 2-142. INTCPS_ILR103 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-149. INTCPS_ILR103 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.143 INTCPS_ILR104 Register (offset = 2A0h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR104 is shown in Figure 2-143 and described in Table 2-150.

Figure 2-143. INTCPS_ILR104 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-150. INTCPS_ILR104 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.144 INTCPS_ILR105 Register (offset = 2A4h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR105 is shown in Figure 2-144 and described in Table 2-151.

Figure 2-144. INTCPS_ILR105 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-151. INTCPS_ILR105 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.145 INTCPS_SIR_FIQ Register (offset = 44h) [reset = 0h]

This register supplies the currently active FIQ interrupt number

INTCPS_SIR_FIQ is shown in Figure 2-145 and described in Table 2-152.

Figure 2-145. INTCPS_SIR_FIQ Register

31	30	29	28	27	26	25	24
SpuriousFIQ							
R							

23	22	21	20	19	18	17	16
SpuriousFIQ							
R							

15	14	13	12	11	10	9	8
SpuriousFIQ							
R							

7	6	5	4	3	2	1	0
SpuriousFIQ	ActiveFIQ						
R	R						

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-152. INTCPS_SIR_FIQ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	SpuriousFIQ	R		Spurious FIQ flag
6-0	ActiveFIQ	R		Active FIQ number

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.146 INTCPS_ILR106 Register (offset = 2A8h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR106 is shown in Figure 2-146 and described in Table 2-153.

Figure 2-146. INTCPS_ILR106 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-153. INTCPS_ILR106 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.147 INTCPS_ILR107 Register (offset = 2ACh) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR107 is shown in Figure 2-147 and described in Table 2-154.

Figure 2-147. INTCPS_ILR107 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-154. INTCPS_ILR107 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.148 INTCPS_ILR108 Register (offset = 2B0h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR108 is shown in Figure 2-148 and described in Table 2-155.

Figure 2-148. INTCPS_ILR108 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-155. INTCPS_ILR108 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.149 INTCPS_ILR109 Register (offset = 2B4h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR109 is shown in Figure 2-149 and described in Table 2-156.

Figure 2-149. INTCPS_ILR109 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-156. INTCPS_ILR109 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.150 INTCPS_ILR110 Register (offset = 2B8h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR110 is shown in Figure 2-150 and described in Table 2-157.

Figure 2-150. INTCPS_ILR110 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-157. INTCPS_ILR110 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.151 INTCPS_ILR111 Register (offset = 2BCh) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR111 is shown in Figure 2-151 and described in Table 2-158.

Figure 2-151. INTCPS_ILR111 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-158. INTCPS_ILR111 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.152 INTCPS_ILR112 Register (offset = 2C0h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR112 is shown in Figure 2-152 and described in Table 2-159.

Figure 2-152. INTCPS_ILR112 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-159. INTCPS_ILR112 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.153 INTCPS_ILR113 Register (offset = 2C4h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR113 is shown in Figure 2-153 and described in Table 2-160.

Figure 2-153. INTCPS_ILR113 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-160. INTCPS_ILR113 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.154 INTCPS_ILR114 Register (offset = 2C8h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR114 is shown in Figure 2-154 and described in Table 2-161.

Figure 2-154. INTCPS_ILR114 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-161. INTCPS_ILR114 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.155 INTCPS_ILR115 Register (offset = 2CCh) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR115 is shown in Figure 2-155 and described in Table 2-162.

Figure 2-155. INTCPS_ILR115 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-162. INTCPS_ILR115 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.156 INTCPS_CONTROL Register (offset = 48h) [reset = 0h]

This register contains the new interrupt agreement bits

INTCPS_CONTROL is shown in Figure 2-156 and described in Table 2-163.

Figure 2-156. INTCPS_CONTROL Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved						NewFIQAgr	NewIRQAgr
R						W	W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-163. INTCPS_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	Reserved	R		Write 0's for future compatibility. Reads returns 0
1	NewFIQAgr	W		Reset FIQ output and enable new FIQ generation 0 - nofun_w: no function effect 1 - NewFiq_w: Reset FIQ output and enable new FIQ generation
0	NewIRQAgr	W		New IRQ generation 0 - nofun_w: no function effect 1 - NewIrq_w: Reset IRQ output and enable new IRQ generation

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.157 INTCPS_ILR116 Register (offset = 2D0h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR116 is shown in Figure 2-157 and described in Table 2-164.

Figure 2-157. INTCPS_ILR116 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-164. INTCPS_ILR116 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.158 INTCPS_ILR117 Register (offset = 2D4h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR117 is shown in Figure 2-158 and described in Table 2-165.

Figure 2-158. INTCPS_ILR117 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-165. INTCPS_ILR117 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.159 INTCPS_ILR118 Register (offset = 2D8h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR118 is shown in Figure 2-159 and described in Table 2-166.

Figure 2-159. INTCPS_ILR118 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-166. INTCPS_ILR118 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.160 INTCPS_ILR119 Register (offset = 2DCh) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR119 is shown in Figure 2-160 and described in Table 2-167.

Figure 2-160. INTCPS_ILR119 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-167. INTCPS_ILR119 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.161 INTCPS_ILR120 Register (offset = 2E0h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR120 is shown in Figure 2-161 and described in Table 2-168.

Figure 2-161. INTCPS_ILR120 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-168. INTCPS_ILR120 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.162 INTCPS_ILR121 Register (offset = 2E4h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR121 is shown in Figure 2-162 and described in Table 2-169.

Figure 2-162. INTCPS_ILR121 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-169. INTCPS_ILR121 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.163 INTCPS_ILR122 Register (offset = 2E8h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR122 is shown in Figure 2-163 and described in Table 2-170.

Figure 2-163. INTCPS_ILR122 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-170. INTCPS_ILR122 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.164 INTCPS_ILR123 Register (offset = 2ECh) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR123 is shown in Figure 2-164 and described in Table 2-171.

Figure 2-164. INTCPS_ILR123 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-171. INTCPS_ILR123 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.165 INTCPS_ILR124 Register (offset = 2F0h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR124 is shown in Figure 2-165 and described in Table 2-172.

Figure 2-165. INTCPS_ILR124 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-172. INTCPS_ILR124 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.166 INTCPS_ILR125 Register (offset = 2F4h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR125 is shown in Figure 2-166 and described in Table 2-173.

Figure 2-166. INTCPS_ILR125 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-173. INTCPS_ILR125 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.167 INTCPS_PROTECTION Register (offset = 4Ch) [reset = 0h]

This register controls protection of the other registers. This register can only be accessed in priviledged mode, regardless of the curent value of the protection bit.

INTCPS_PROTECTION is shown in Figure 2-167 and described in Table 2-174.

Figure 2-167. INTCPS_PROTECTION Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							Protection
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-174. INTCPS_PROTECTION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		Write 0's for future compatibility. Reads returns 0
0	Protection	R/W		Protection mode 0 - ProtMDis: Protection mode disabled (default) 1 - ProtMEnb: Protection mode enabled. When enabled, only priviledged mode can access registers.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.168 INTCPS_ILR126 Register (offset = 2F8h) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR126 is shown in Figure 2-168 and described in Table 2-175.

Figure 2-168. INTCPS_ILR126 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-175. INTCPS_ILR126 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.169 INTCPS_ILR127 Register (offset = 2FCh) [reset = 0h]

These registers contain the priority for the interrupts and the FIQ/IRQ steering.

INTCPS_ILR127 is shown in Figure 2-169 and described in Table 2-176.

Figure 2-169. INTCPS_ILR127 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							Priority
R							R/W

7	6	5	4	3	2	1	0
Priority						Reserved	FIQnIRQ
R/W							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-176. INTCPS_ILR127 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		Write 0's for future compatibility. Reads returns 0
8-2	Priority	R/W		Interrupt priority
0	FIQnIRQ	R/W		Interrupt IRQ FiQ mapping 0 - IntIRQ: Interrupt is routed to IRQ 1 - IntFIQ: Interrupt is routed to FIQ

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.170 INTCPS_IDLE Register (offset = 50h) [reset = 0h]

This register controls the clock auto-idle for the functional clock and the input synchronisers

INTCPS_IDLE is shown in Figure 2-170 and described in Table 2-177.

Figure 2-170. INTCPS_IDLE Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved						Turbo	FuncIdle
R						R/W	R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-177. INTCPS_IDLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	Reserved	R		Write 0's for future compatibility. Reads returns 0
1	Turbo	R/W		Input synchroniser clock auto-gating 0 - SyncFree: Input synchroniser clock is free running (default) 1 - SyncAuto: Input synchroniser clock is auto-gated based on interrupt input activity
0	FuncIdle	R/W		Functional clock auto-idle mode 0 - FuncAuto: Functional clock gating strategy is applied (default) 1 - FuncFree: Functional clock is free-running

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

2.7.171 INTCPS_IRQ_PRIORITY Register (offset = 60h) [reset = 0h]

This register supplies the currently active IRQ priority level

INTCPS_IRQ_PRIORITY is shown in Figure 2-171 and described in Table 2-178.

Figure 2-171. INTCPS_IRQ_PRIORITY Register

31	30	29	28	27	26	25	24
SpuriousIRQflag							
R							

23	22	21	20	19	18	17	16
SpuriousIRQflag							
R							

15	14	13	12	11	10	9	8
SpuriousIRQflag							
R							

7	6	5	4	3	2	1	0
SpuriousIRQflag	IRQPriority						
R	R						

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 2-178. INTCPS_IRQ_PRIORITY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	SpuriousIRQflag	R		Spurious IRQ flag
6-0	IRQPriority	R		Current IRQ priority

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Memory Map

This sections describes the memory map for the device.

3.1 ARM Cortex A8 Memory Map

Table 3-1. L3 Memory Map

Block Name	Start_address (hex)	End_address (hex)	Size	Description
GPMC	0x0000_0000	0x1FFF_FFFF	512MB	8/16 Ex/R/W[(1)]
Reserved	0x2000_0000	0x3FFF_FFFF	512MB	Reserved
Boot ROM internal[(2)]	0x4000_0000	0x4001_FFFF	128KB	32-bit Ex/R[(1)] – Secure
	0x4002_0000	0x4002_BFFF	48KB	32-bit Ex/R[(1)] – Public
Reserved[(2)] (Boot ROM)	0x4002_C000	0x400F_FFFF	848KB	Reserved (Boot ROM)
Reserved[(2)]	0x4010_0000	0x401F_FFFF	1MB	Reserved
Reserved[(2)]	0x4020_0000	0x402E_FFFF	960KB	Reserved
SRAM internal[(2)]	0x402F_0000	0x402F_FFFF	64KB	32-bit Ex/R/W[(1)] – Secure SRAM
L3 OCMC0	0x4030_0000	0x4030_FFFF	64KB	32-bit Ex/R/W[(1)] OCMC SRAM
Reserved	0x4031_0000	0x403F_FFFF	960KB	Reserved
Reserved	0x4040_0000	0x4041_FFFF	128KB	Reserved
Reserved	0x4042_0000	0x404F_FFFF	896KB	Reserved
Reserved	0x4050_0000	0x405F_FFFF	1MB	Reserved
Reserved	0x4060_0000	0x407F_FFFF	2MB	Reserved
Reserved	0x4080_0000	0x4083_FFFF	256KB	Reserved
Reserved	0x4084_0000	0x40DF_FFFF	5888KB	Reserved
Reserved	0x40E0_0000	0x40E0_7FFF	32KB	Reserved
Reserved	0x40E0_8000	0x40EF_FFFF	992KB	Reserved
Reserved	0x40F0_0000	0x40F0_7FFF	32KB	Reserved
Reserved	0x40F0_8000	0x40FF_FFFF	992KB	Reserved
Reserved	0x4100_0000	0x41FF_FFFF	16MB	Reserved
Reserved	0x4200_0000	0x43FF_FFFF	32MB	Reserved
L3F CFG Regs	0x4400_0000	0x443F_FFFF	4MB	L3Fast configuration registers
Reserved	0x4440_0000	0x447F_FFFF	4MB	Reserved
L3S CFG Regs	0x4480_0000	0x44BF_FFFF	4MB	L3Slow configuration registers
L4_WKUP	0x44C0_0000	0x44FF_FFFF	4MB	L4_WKUP
Expansion Port	0x4500_0000	0x45FF_FFFF	16MB	Expansion L3 port
McASP0	0x4600_0000	0x463F_FFFF	4MB	McASP0
McASP1	0x4640_0000	0x467F_FFFF	4MB	McASP1
Reserved	0x4680_0000	0x46FF_FFFF	8MB	Reserved

(1) Ex/R/W – Execute/Read/Write.
(2) These addresses are decoded within the Sabertooth subsystem

Copyright © 2011, Texas Instruments Incorporated

Table 3-1. L3 Memory Map (continued)

Block Name	Start_address (hex)	End_address (hex)	Size	Description
Reserved	0x4700_0000	0x473F_FFFF	4MB	Reserved
USB	0x4740_0000	0x477F_FFFF	4MB	USB Registers / CPPI
Reserved	0x4780_0000	0x4780_FFFF	64KB	Reserved
MMCHS2	0x4781_0000	0x4781_FFFF	64KB	MMCHS2
Reserved	0x4782_0000	0x47BF_FFFF	4MB-128KB	Reserved
L4_FW	0x47C0_0000	0x47FF_FFFF	4MB	L4 Firewall Configuration (see L4_FW table)
L4_PER	0x4800_0000	0x48FF_FFFF	16MB	L4 Peripheral (see L4_PER table)
TPCC	0x4900_0000	0x490F_FFFF	1MB	TPCC Registers
Reserved	0x4910_0000	0x497F_FFFF	7MB	Reserved
TPTC0	0x4980_0000	0x498F_FFFF	1MB	TPTC0 Registers
TPTC1	0x4990_0000	0x499F_FFFF	1MB	TPTC1 Registers
TPTC2	0x49A0_0000	0x49AF_FFFF	1MB	TPTC2 Registers
Reserved	0x49B0_0000	0x49BF_FFFF	1MB	Reserved
Reserved	0x49C0_0000	0x49FF_FFFF	4MB	Reserved (TPTC4-7)
L4_FAST	0x4A00_0000	0x4AFF_FFFF	16MB	L4_FAST
DebugSS	0x4B00_0000	0x4BFF_FFFF	16MB	Debug Subsystem region
EMIF0 registers	0x4C00_0000	0x4CFF_FFFF	16MB	EMIF0 Configuration registers
Reserved	0x4D00_0000	0x4DFF_FFFF	16MB	Reserved
Reserved	0x4E00_0000	0x4FFF_FFFF	32MB	Reserved
GPMC Registers	0x5000_0000	0x50FF_FFFF	16MB	GPMC Configuration registers
Reserved	0x5100_0000	0x52FF_FFFF	32MB	Reserved
SHA	0x5300_0000	0x530F_FFFF	1MB	Module - (Maddrspace 0)
	0x5310_0000	0x531F_FFFF	1MB	Module - (Maddrspace 1)
Reserved	0x5320_0000	0x533F_FFFF	2MB	Reserved
AES0	0x5340_0000	0x534F_FFFF	1MB	Module - (Maddrspace 0)
	0x5350_0000	0x535F_FFFF	1MB	Module - (Maddrspace 1)
Reserved	0x5360_0000	0x54BF_FFFF	22MB	Reserved
ADC_TSC	0x54C0_0000	0x54FF_FFFF	4MB	ADC_TSC DMA Port
Reserved	0x5500_0000	0x55FF_FFFF	16MB	Reserved
SGX530	0x5600_0000	0x56FF_FFFF	16MB	SGX530 Slave Port
Reserved	0x5700_0000	0x57FF_FFFF	16MB	Reserved
Reserved	0x5800_0000	0x58FF_FFFF	16MB	Reserved
Reserved	0x5900_0000	0x59FF_FFFF	16MB	Reserved
Reserved	0x5A00_0000	0x5AFF_FFFF	16MB	Reserved
Reserved	0x5B00_0000	0x5BFF_FFFF	16MB	Reserved
Reserved	0x5C00_0000	0x5DFF_FFFF	32MB	Reserved
Reserved	0x5E00_0000	0x5FFF_FFFF	32MB	Reserved
Reserved	0x6000_0000	0x7FFF_FFFF	512MB	Reserved
EMIF0 SDRAM	0x8000_0000	0xBFFF_FFFF	1GB	DDR (1 GB Space)
Reserved	0xC000_0000	0x1_FFFF_FFFF	5GB	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

3.2 ARM Cortex M3 Memory Map

Table 3-2. M3 Processor Memory Map

Region Name	Start Address (hex)	End Address (hex)	Size	Description
M3 Unified Memory (UMEM)	0x0000_0000	0x0000_3FFF	16 KB	ICODE & DCODE unified code space (ICode, DCode Buses are muxed into single Memory: UMEM)
Reserved	0x0000_4000	0x0007_FFFF	512KB-16KB	Reserved
M3 DMEM	0x0008_0000	0x0008_1FFF	8KB	M3 Data Memory
Reserved	0x0008_2000	0x000F_FFFF	512KB-8KB	Reserved
Reserved	0x0010_0000	0x1FFF_FFFF	511MB	Reserved
M3 Unified Memory[(1)] (UMEM)	0x2000_0000	0x2000_3FFF	16KB	Through L4_WKUP (using System Bus) (SRAM Bit Band Region)
Reserved	0x2000_4000	0x2007_FFFF	512KB-16KB	Reserved (SRAM Bit Band Region)
M3 DMEM[(1)]	0x2008_0000	0x2008_1FFF	8KB	Through L4_WKUP (using System Bus) (SRAM Bit Band Region)
Reserved[(2)]	0x2008_2000	0x200F_FFFF	512KB-8KB	Reserved (SRAM Bit Band Region)
Reserved[(2)]	0x2010_0000	0x21FF_FFFF	31MB	Reserved (SRAM Region)
Bit-Band Alias Region[(3)]	0x2200_0000	0x23FF_FFFF	32MB	Bit-Band Alias Region
Reserved[(2)]	0x2400_0000	0x3FFF_FFFF	448MB	Reserved (SRAM Region)
Reserved[(2)]	0x4000_0000	0x400F_FFFF	1MB	Reserved (Periph Bit Band Region)
Reserved[(2)]	0x4010_0000	0x41FF_FFFF	31MB	Reserved (Periph Region)
Reserved[(2) (3)]	0x4200_0000	0x43FF_FFFF	32MB	Reserved (Periph Bit-Band Alias)
Reserved[(2)]	0x4400_0000	0x44BF_FFFF	12MB	Reserved (Peripheral Region)
L4_WKUP[(4) (1)]	0x44C0_0000	0x44FF_FFFF	4MB	L4 Wake Up Domain (see Table 3-5)
Reserved[(2)]	0x4500_0000	0x5FFF_FFFF	432MB	Reserved (Peripheral Region)
Reserved[(2)]	0x6000_0000	0xDFFF_FFFF	2GB	Reserved (External RAM/Device)
Reserved (ITM)	0xE000_0000	0xE000_0FFF	4KB	Instrumentation Trace Macrocell
Reserved (DWT)	0xE000_1000	0xE000_1FFF	4KB	Data Watchpoint Trace
Reserved (FPB)	0xE000_2000	0xE000_2FFF	4KB	Flashpatch and Breakpoint
Reserved (Private Peripheral Bus –Internal)	0xE000_3000	0xE000_DFFF	44KB	Reserved (Private Peripheral Bus –Internal)
NVIC	0xE000_E000	0xE000_EFFF	4KB	Nested Vector Interrupt Controller

(1) The Wakeup M3 may access the UMEM and DMEM using the L4_WKUP (OCP Bus) or it may use its direct buses, by accessing address 0x00000000 – 0x00081FFF.
(2) These addresses correspond to L3 regions. These are reserved regions with respect to Wakeup M3.
(3) Accessing these bit-band alias regions results in an access to the corresponding bit-band region.
(4) Peripherals located on L4 Wakeup Bus can be accessed by Wakeup M3.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 3-2. M3 Processor Memory Map (continued)

Region Name	Start Address (hex)	End Address (hex)	Size	Description
Reserved	0xE000_F000	0xE003_FFFF	196KB	Reserved
Reserved	0xE004_0000	0xE004_0FFF	4KB	Reserved
Reserved	0xE004_1000	0xE004_1FFF	4KB	Reserved
Reserved	0xE004_2000	0xE00F_EFFF	756KB	Reserved
ROM table (registers)	0xE00F_F000	0xE00F_FFFF	4KB	ROM table (registers)
Reserved	0xE010_0000	0xFFFF_FFFF	511MB	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

3.3 L4 Memory Map

Table 3-3. L4 Fast Peripheral Memory Map

Device Name	Start_address (hex)	End_address (hex)	Size	Description
L4_Fast configuration	0x4A00_0000	0x4A00_07FF	2KB	Address/Protection (AP)
	0x4A00_0800	0x4A00_0FFF	2KB	Link Agent (LA)
	0x4A00_1000	0x4A00_13FF	1KB	Initiator Port (IP0)
	0x4A00_1400	0x4A00_17FF	1KB	Reserved
	0x4A00_1800	0x4A00_1FFF	2KB	Reserved (IP2 – IP3)
Reserved	0x4A00_2000	0x4A07_FFFF	504KB	Reserved
Reserved	0x4A08_0000	0x4A09_FFFF	128KB	Reserved
	0x4A0A_0000	0x4A0A_0FFF	4KB	Reserved
Reserved	0x4A0A_1000	0x4A0F_FFFF	380KB	Reserved
GEMAC (2P GEMAC SW)	0x4A10_0000	0x4A10_7FFF	32KB	Registers
	0x4A10_8000	0x4A10_8FFF	4KB	L4 Interconnect
Reserved	0x4A10_9000	0x4A13_FFFF	220KB	Reserved
Reserved	0x4A14_0000	0x4A14_FFFF	64KB	Reserved
	0x4A15_0000	0x4A15_0FFF	4KB	Reserved
Reserved	0x4A15_1000	0x4A17_FFFF	188KB	Reserved
Reserved	0x4a18_0000	0x4A1A1FFF	136KB	Reserved
Reserved	0x4A1A2000	0x4A1A3FFF	8KB	Reserved
	0x4A1A4000	0x4A1A4FFF	4KB	Reserved
Reserved	0x4A1A5000	0x4A1A5FFF	4KB	Reserved
	0x4A1A6000	0x4A1A6FFF	4KB	Reserved
Reserved	0x4A1A7000	0x4A1A7FFF	4KB	Reserved
Reserved	0x4A1A8000	0x4A1A9FFF	8KB	Reserved
	0x4A1AA000	0x4A1AAFFF	4KB	Reserved
Reserved	0x4A1AB000	0x4A1ABFFF	4KB	Reserved
	0x4A1AC000	0x4A1ACFFF	4KB	Reserved
Reserved	0x4A1AD000	0x4A1ADFFF	4KB	Reserved
Reserved	0x4A1AE000	0x4A1AFFFF	8KB	Reserved
	0x4A1B0000	0x4A1B0FFF	4KB	Reserved
Reserved	0x4A1B1000	0x4A1B1FFF	4KB	Reserved
	0x4A1B2000	0x4A1B2FFF	4KB	Reserved
Reserved	0x4A1B3000	0x4A1B3FFF	4KB	Reserved
	0x4A1B4000	0x4A1B4FFF	4KB	Reserved
Reserved	0x4A1B5000	0x4A1B5FFF	4KB	Reserved
	0x4A1B6000	0x4A1B6FFF	4KB	Reserved
Reserved	0x4A1B_4000	0x4A1F_FFFF	304KB	Reserved
Reserved	0x4A20_0000	0x4A2F_FFFF	1MB	Reserved
PRUSS1	0x4A30_0000	0x4A37_FFFF	512KB	PRUSS1 Instruction/Data/Control Space
	0x4A38_0000	0x4A38_0FFF	4KB	L4 Interconnect
Reserved	0x4A38_1000	0x4A3F_FFFF	508KB	Reserved
Reserved	0x4A40_0000	0x4AFF_FFFF	12MB	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 3-4. L4_PER Peripheral Memory Map

Device Name	Start_address (hex)	End_address (hex)	Size	Description
Reserved	0x4800_0000	0x4800_07FF	2KB	Address/Protection (AP)
	0x4800_0800	0x4800_0FFF	2KB	Link Agent (LA)
	0x4800_1000	0x4800_13FF	1KB	Initiator Port (IP0)
	0x4800_1400	0x4800_17FF	1KB	Initiator Port (IP1)
	0x4800_1800	0x4800_1BFF	1KB	Initiator Port (IP2)
	0x4800_1C00	0x4800_1FFF	1KB	Initiator Port (IP3)
Reserved	0x4800_2000	0x4800_3FFF	8KB	Reserved
Reserved	0x4800_4000	0x4800_7FFF	16KB	Reserved
eFuse Ctrl	0x4800_8000	0x4800_8FFF	4KB	Module
	0x4800_9000	0x4800_9FFF	4KB	L4 Interconnect
Reserved	0x4800_A000	0x4800_FFFF	24KB	Reserved
Reserved	0x4801_0000	0x4801_0FFF	4KB	Reserved
	0x4801_1000	0x4801_1FFF	4KB	Reserved
Reserved	0x4801_2000	0x4801_3FFF	8KB	Reserved
Reserved	0x4801_4000	0x4801_FFFF	48KB	Reserved
Reserved[(1)]	0x4802_0000	0x4802_0FFF	4KB	Reserved
	0x4802_1000	0x4802_1FFF	4KB	Reserved
UART1	0x4802_2000	0x4802_2FFF	4KB	Module
	0x4802_3000	0x4802_3FFF	4KB	L4 Interconnect
UART2	0x4802_4000	0x4802_4FFF	4KB	Module
	0x4802_5000	0x4802_5FFF	4KB	L4 Interconnect
Reserved	0x4802_6000	0x4802_7FFF	8KB	Reserved
Reserved[(1)]	0x4802_8000	0x4802_8FFF	4KB	Reserved
	0x4802_9000	0x4802_9FFF	4KB	Reserved
I2C1	0x4802_A000	0x4802_AFFF	4KB	Module
	0x4802_B000	0x4802_BFFF	4KB	L4 Interconnect
Reserved[(1)]	0x4802_C000	0x4802_CFFF	4KB	Reserved
	0x4802_D000	0x4802_DFFF	4KB	Reserved
Reserved[(1)]	0x4802_E000	0x4802_EFFF	4KB	Reserved
	0x4802_F000	0x4802_FFFF	4KB	Reserved
MCSPIOCP 0	0x4803_0000	0x4803_0FFF	4KB	Module
	0x4803_1000	0x4803_1FFF	4KB	L4 Interconnect
Reserved[(1)]	0x4803_2000	0x4803_2FFF	4KB	Reserved
	0x4803_3000	0x4803_3FFF	4KB	Reserved
Reserved	0x4803_4000	0x4803_4FFF	4KB	Reserved
	0x4803_5000	0x4803_5FFF	4KB	Reserved
Reserved	0x4803_6000	0x4803_6FFF	4KB	Reserved
	0x4803_7000	0x4803_7FFF	4KB	Reserved
McASP0 CFG	0x4803_8000	0x4803_9FFF	8KB	Module
	0x4803_A000	0x4803_AFFF	4KB	L4 Interconnect
Reserved	0x4803_B000	0x4803_BFFF	4KB	Reserved
McASP1 CFG	0x4803_C000	0x4803_DFFF	8KB	Module
	0x4803_E000	0x4803_EFFF	4KB	L4 Interconnect
Reserved	0x4803_F000	0x4803_FFFF	4KB	Reserved
DMTIMER2	0x4804_0000	0x4804_0FFF	4KB	Module
	0x4804_1000	0x4804_1FFF	4KB	L4 Interconnect
DMTIMER3	0x4804_2000	0x4804_2FFF	4KB	Module

(1) On AM335x these modules are part of L4_WKUP.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 3-4. L4_PER Peripheral Memory Map (continued)

Device Name	Start_address (hex)	End_address (hex)	Size	Description
	0x4804_3000	0x4804_3FFF	4KB	L4 Interconnect
DMTIMER4	0x4804_4000	0x4804_4FFF	4KB	Module
	0x4804_5000	0x4804_5FFF	4KB	L4 Interconnect
DMTIMER5	0x4804_6000	0x4804_6FFF	4KB	Module
	0x4804_7000	0x4804_7FFF	4KB	L4 Interconnect
DMTIMER6	0x4804_8000	0x4804_8FFF	4KB	Module
	0x4804_9000	0x4804_9FFF	4KB	L4 Interconnect
DMTIMER7	0x4804_A000	0x4804_AFFF	4KB	Module
	0x4804_B000	0x4804_BFFF	4KB	L4 Interconnect
GPIO1	0x4804_C000	0x4804_CFFF	4KB	Module
	0x4804_D000	0x4804_DFFF	4KB	L4 Interconnect
Reserved	0x4804_E000	0x4804_FFFF	8KB	Reserved
Reserved	0x4805_0000	0x4805_FFFF	64KB	Reserved
MMCHS 0	0x4806_0000	0x4806_0FFF	4KB	Registers
	0x4806_1000	0x4806_1FFF	4KB	L4 Interconnect
Reserved	0x4806_2000	0x4807_FFFF	120KB	Reserved
ELM	0x4808_0000	0x4808_FFFF	64KB	Module
	0x4809_0000	0x4809_0FFF	4KB	L4 Interconnect
Reserved	0x4809_1000	0x4809_FFFF	60KB	Reserved
Spare0	0x480A_0000	0x480A_FFFF	64KB	Spare0 Module
	0x480B_0000	0x480B_0FFF	4KB	L4 Interconnect
Reserved	0x480B_1000	0x480B_FFFF	60KB	Reserved
Reserved[(2)]	0x480C_0000	0x480C_0FFF	4KB	Reserved
	0x480C_1000	0x480C_1FFF	4KB	Reserved
Reserved[(2)]	0x480C_2000	0x480C_2FFF	4KB	Reserved
	0x480C_3000	0x480C_3FFF	4KB	Reserved
Reserved	0x480C_4000	0x480C_7FFF	16KB	Reserved
Mailbox 0	0x480C_8000	0x480C_8FFF	4KB	Module
	0x480C_9000	0x480C_9FFF	4KB	L4 Interconnect
Spinlock	0x480C_A000	0x480C_AFFF	4KB	Module
	0x480C_B000	0x480C_BFFF	4KB	L4 Interconnect
Reserved	0x480C_C000	0x480F_FFFF	208KB	Reserved
Reserved	0x4810_0000	0x4811_FFFF	128KB	Reserved
	0x4812_0000	0x4812_0FFF	4KB	Reserved
Reserved	0x4812_1000	0x4812_1FFF	4KB	Reserved
Reserved	0x4812_2000	0x4812_2FFF	4KB	Reserved
	0x4812_3000	0x4812_3FFF	4KB	Reserved
Reserved	0x4812_4000	0x4813_FFFF	112KB	Reserved
Reserved[(2)]	0x4814_0000	0x4815_FFFF	128KB	Reserved
	0x4816_0000	0x4816_0FFF	4K	Reserved
Reserved	0x4816_1000	0x4817_FFFF	124KB	Reserved
Reserved[(2)]	0x4818_0000	0x4818_2FFF	12KB	Reserved
	0x4818_3000	0x4818_3FFF	4KB	Reserved
Reserved	0x4818_4000	0x4818_7FFF	16KB	Reserved
Reserved[(2)]	0x4818_8000	0x4818_8FFF	4KB	Reserved
	0x4818_9000	0x4818_9FFF	4KB	Reserved

(2) On AM335x these modules are part of L4_WKUP.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 3-4. L4_PER Peripheral Memory Map (continued)

Device Name	Start_address (hex)	End_address (hex)	Size	Description
Reserved[(2)]	0x4818_A000	0x4818_AFFF	4KB	Reserved
	0x4818_B000	0x4818_BFFF	4KB	Reserved
OCP Watchpoint	0x4818_C000	0x4818_CFFF	4KB	Module
	0x4818_D000	0x4818_DFFF	4KB	L4 Interconnect
P1500 CFG (IEEE1500)	0x4818_E000	0x4818_EFFF	4KB	Module
	0x4818_F000	0x4818_FFFF	4KB	L4 Interconnect
Reserved	0x4819_0000	0x4819_0FFF	4KB	Reserved
	0x4819_1000	0x4819_1FFF	4KB	Reserved
Reserved	0x4819_2000	0x4819_2FFF	4KB	Reserved
	0x4819_3000	0x4819_3FFF	4KB	Reserved
Reserved	0x4819_4000	0x4819_BFFF	32KB	Reserved
I2C2	0x4819_C000	0x4819_CFFF	4KB	Module
	0x4819_D000	0x4819_DFFF	4KB	L4 Interconnect
Reserved	0x4819_E000	0x4819_EFFF	4KB	Reserved
	0x4819_F000	0x4819_FFFF	4KB	Reserved
MCSPIOCP1	0x481A_0000	0x481A_0FFF	4KB	Module
	0x481A_1000	0x481A_1FFF	4KB	L4 Interconnect
Reserved	0x481A_2000	0x481A_5FFF	16KB	Reserved
UART3	0x481A_6000	0x481A_6FFF	4KB	Module
	0x481A_7000	0x481A_7FFF	4KB	L4 Interconnect
UART4	0x481A_8000	0x481A_8FFF	4KB	Module
	0x481A_9000	0x481A_9FFF	4KB	L4 Interconnect
UART5	0x481A_A000	0x481A_AFFF	4KB	Module
	0x481A_B000	0x481A_BFFF	4KB	L4 Interconnect
GPIO 2	0x481A_C000	0x481A_CFFF	4KB	Module
	0x481A_D000	0x481A_DFFF	4KB	L4 Interconnect
GPIO 3	0x481A_E000	0x481A_EFFF	4KB	Module
	0x481A_F000	0x481A_FFFF	4KB	L4 Interconnect
Spare 1	0x481B_0000	0x481B_FFFF	64KB	Module
	0x481C_0000	0x481C_0FFF	4KB	L4 Interconnect
Reserved	0x481C_1000	0x481C_1FFF	4KB	Reserved
	0x481C_2000	0x481C_2FFF	4KB	Reserved
Reserved	0x481C_3000	0x481C_9FFF	28KB	Reserved
Reserved	0x481C_A000	0x481C_AFFF	4KB	Reserved
	0x481C_B000	0x481C_BFFF	4KB	Reserved
DCAN 0	0x481C_C000	0x481C_DFFF	8KB	Module
	0x481C_E000	0x481C_FFFF	8KB	L4 Interconnect
DCAN 1	0x481D_0000	0x481D_1FFF	8KB	Module
	0x481D_2000	0x481D_3FFF	8KB	L4 Interconnect
Reserved	0x481D_4000	0x481D_4FFF	4KB	Reserved
	0x481D_5000	0x481D_5FFF	4KB	Reserved
Reserved	0x481D_6000	0x481D_6FFF	4KB	Reserved
	0x481D_7000	0x481D_7FFF	4KB	Reserved
MMCHS 1	0x481D_8000	0x481D_8FFF	4KB	Module
	0x481D_9000	0x481D_9FFF	4KB	L4 Interconnect
Reserved	0x481D_A000	0x481F_FFFF	152KB	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 3-4. L4_PER Peripheral Memory Map (continued)

Device Name	Start_address (hex)	End_address (hex)	Size	Description
Interrupt controller[3]	0x4820_0000	0x4820_0FFF	4KB	Host ARM non shared device mapping
Reserved[3]	0x4820_1000	0x4823_FFFF	252KB	Host ARM non shared device mapping
MPUSS config register[3]	0x4824_0000	0x4824_0FFF	4KB	Host ARM non shared device mapping
Reserved[3]	0x4824_1000	0x4827_FFFF	252KB	Host ARM non shared device mapping
Reserved[3]	0x4828_0000	0x4828_0FFF	4KB	Host ARM non shared device mapping
Reserved[3]	0x4828_1000	0x482F_FFFF	508KB	Host ARM non shared device mapping
eCAP/eQEP/ePWM0	0x4830_0000	0x4830_0FFF	4KB	Module
	0x4830_1000	0x4830_1FFF	4KB	L4 Interconnect
eCAP/eQEP/ePWM1	0x4830_2000	0x4830_2FFF	4KB	Module
	0x4830_3000	0x4830_3FFF	4KB	L4 Interconnect
eCAP/eQEP/ePWM2	0x4830_4000	0x4830_4FFF	4KB	Module
	0x4830_5000	0x4830_5FFF	4KB	L4 Interconnect
Reserved	0x4830_6000	0x4830_DFFF	32KB	Reserved
LCDC	0x4830_E000	0x4830_EFFF	4KB	Module
	0x4830_F000	0x4830_FFFF	4KB	L4 Interconnect
RNG	0x4831_0000	0x4831_1FFF	8KB	Module
	0x4831_2000	0x4831_2FFF	4KB	L4 Interconnect
Reserved	0x4831_3000	0x4831_7FFF	20KB	Reserved
PKA	0x4831_8000	0x4831_BFFF	16KB	Module
	0x4831_C000	0x4831_CFFF	4KB	L4 Interconnect
Reserved	0x4831_D000	0x4831_FFFF	12KB	L4 Interconnect
Reserved	0x4832_0000	0x4832_5FFF	16KB	Reserved
Reserved	0x4832_6000	0x48FF_FFFF	13MB-152KB	Reserved

(3) These regions are decoded internally by the Host ARM Subsystem and are not physically part of the L4 Slow. For masters other than the Host ARM these regions are reserved.

Table 3-5. L4_FW Firewall Configuration Memory Map

Device Name	Start_address (hex)	End_address (hex)	Size	Description
L4_Firewall configuration	0x47C0_0000	0x47C0_07FF	2KB	Address/Protection (AP)
	0x47C0_0800	0x47C0_0FFF	2KB	Link Agent (LA)
	0x47C0_1000	0x47C0_13FF	1KB	Initiator Port (IP0)
	0x47C0_1400	0x47C0_1FFF	3KB	Reserved (IP1-IP3)
Reserved	0x47C0_2000	0x47C0_5FFF	16KB	Reserved
Reserved	0x47C0_6000	0x47C0_7FFF	8KB	Reserved
Reserved	0x47C0_8000	0x47C0_BFFF	16KB	Reserved
EMIF Firewall	0x47C0_C000	0x47C0_CFFF	4KB	Module
	0x47C0_D000	0x47C0_DFFF	4KB	L4 interconnect
GPMC Firewall	0x47C0_E000	0x47C0_EFFF	4KB	Module
	0x47C0_F000	0x47C0_FFFF	4KB	L4 interconnect
OCMC RAM Firewall	0x47C1_0000	0x47C1_0FFF	4KB	Module
	0x47C1_1000	0x47C1_1FFF	4KB	L4 interconnect
Reserved	0x47C1_2000	0x47C1_3FFF	8KB	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 3-5. L4_FW Firewall Configuration Memory Map (continued)

Device Name	Start_address (hex)	End_address (hex)	Size	Description
SGX530-T Firewall	0x47C1_4000	0x47C1_4FFF	4KB	Module
	0x47C1_5000	0x47C1_5FFF	4KB	L4 interconnect
Reserved	0x47C1_6000	0x47C1_6FFF	4KB	Reserved
	0x47C1_7000	0x47C1_7FFF	4KB	Reserved
Reserved	0x47C1_8000	0x47C1_8FFF	4KB	Reserved
	0x47C1_9000	0x47C1_9FFF	4KB	Reserved
Reserved	0x47C1_A000	0x47C1_CFFF	12KB	Reserved
	0x47C1_D000	0x47C1_DFFF	4KB	Reserved
Reserved	0x47C1_E000	0x47C1_EFFF	4KB	Reserved
	0x47C1_F000	0x47C1_FFFF	4KB	Reserved
Reserved	0x47C2_0000	0x47C2_0FFF	4KB	Reserved
	0x47C2_1000	0x47C2_1FFF	4KB	Reserved
Reserved	0x47C2_2000	0x47C2_2FFF	4KB	Reserved
	0x47C2_3000	0x47C2_3FFF	4KB	Reserved
MMCHS2 Firewall	0x47C2_4000	0x47C2_4FFF	4KB	Module
	0x47C2_5000	0x47C2_5FFF	4KB	L4 interconnect
McASP0 Firewall	0x47C2_6000	0x47C2_6FFF	4KB	Module
	0x47C2_7000	0x47C2_7FFF	4KB	L4 interconnect
McASP1 Firewall	0x47C2_8000	0x47C2_8FFF	4KB	Module
	0x47C2_9000	0x47C2_9FFF	4KB	L4 interconnect
Reserved	0x47C2_A000	0x47C2_AFFF	4KB	Reserved
	0x47C2_B000	0x47C2_BFFF	4KB	Reserved
Reserved	0x47C2_C000	0x47C2_CFFF	4KB	Reserved
	0x47C2_D000	0x47C2_DFFF	4KB	Reserved
Reserved	0x47C2_E000	0x47C2_EFFF	4KB	Reserved
	0x47C2_F000	0x47C2_FFFF	4KB	Reserved
TPTC Firewall	0x47C3_0000	0x47C3_0FFF	4KB	Module
	0x47C3_1000	0x47C3_1FFF	4KB	L4 interconnect
Reserved	0x47C3_2000	0x47C3_7FFF	24KB	Reserved
TPCC Firewall	0x47C3_8000	0x47C3_8FFF	4KB	Module
	0x47C3_9000	0x47C3_9FFF	4KB	L4 interconnect
USB Firewall	0x47C3_A000	0x47C3_AFFF	4KB	Module
	0x47C3_B000	0x47C3_BFFF	4KB	L4 Interconnect
Expansion Slot Firewall	0x47C3_C000	0x47C3_CFFF	4KB	Module
	0x47C3_D000	0x47C3_DFFF	4KB	L4 interconnect
DebugSS L3 Firewall	0x47C3_E000	0x47C3_EFFF	4KB	Module
	0x47C3_F000	0x47C3_FFFF	4KB	L4 interconnect
AES0 Firewall	0x47C4_0000	0x47C4_0FFF	4KB	Module
	0x47C4_1000	0x47C4_1FFF	4KB	L4 Interconnect
Reserved	0x47C4_2000	0x47C4_3FFF	8KB	Reserved
SHA Firewall	0x47C4_4000	0x47C4_4FFF	4KB	Module
	0x47C4_5000	0x47C4_5FFF	4KB	L4 Interconnect
ADC_TSC Firewall	0x47C4_6000	0x47C4_6FFF	4KB	Module
	0x47C4_7000	0x47C4_7FFF	4KB	L4 Interconnect
Reserved	0x47C4_8000	0x47FF_FFFF	4MB-288KB	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 3-6. L4_WKUP Peripheral Memory Map

Region Name	Start Address (hex)	End Address (hex)	Size	Description
Reserved L3 Slave Region[(1) (2)]	0x0000_0000	0x44BF_FFFF	1G+76MB	Reserved
(L4_WKUP[(3)])	(0x44C0_0000)	(0x44FF_FFFF)	(4MB)	(L4 Wake Up Region)
L4_WKUP configuration	0x44C0_0000	0x44C0_07FF	2KB	Address/Protection (AP)
	0x44C0_0800	0x44C0_0FFF	2KB	Link Agent (LA)
	0x44C0_1000	0x44C0_13FF	1KB	Initiator Port (IP0)
	0x44C0_1400	0x44C0_17FF	1KB	Initiator Port (IP1)
Reserved	0x44C0_1800	0x44C0_1FFF	2KB	Reserved (IP2 – IP3)
Reserved[(3)]	0x44C0_2000	0x44CF_FFFF	1MB-8KB	Reserved
M3 UMEM[(4) (3)]	0x44D0_0000	0x44D0_3FFF	16KB	M3 Unified Code Space (M3 Icode & Dcode buses can access this RAM)
	0x44D0_4000	0x44D0_4FFF	4KB	L4 Interconnect
Reserved	0x44D0_5000	0x44D7_FFFF	492KB	Reserved
M3 DMEM[(4) (3)]	0x44D8_0000	0x44D8_1FFF	8KB	M3 Data Memory
	0x44D8_2000	0x44D8_2FFF	4KB	L4 Interconnect
Reserved	0x44D8_3000	0x44DF_FFFF	500KB	Reserved
PRCM[(3)]	0x44E0_0000	0x44E0_1FFF	8KB	Module
	0x44E0_2000	0x44E0_2FFF	4KB	L4 Interconnect
Spare[(3)] (Previously Clock Manager)	0x44E0_3000	0x44E0_3FFF	4KB	Module
	0x44E0_4000	0x44E0_4FFF	4KB	L4 Interconnect
DMTIMER0[(3)] (Secure Timer)	0x44E0_5000	0x44E0_5FFF	4KB	Module
	0x44E0_6000	0x44E0_6FFF	4KB	L4 Interconnect
GPIO0[(3)]	0x44E0_7000	0x44E0_7FFF	4KB	Module
	0x44E0_8000	0x44E0_8FFF	4KB	L4 Interconnect
UART0[(3)]	0x44E0_9000	0x44E0_9FFF	4KB	Module
	0x44E0_A000	0x44E0_AFFF	4KB	L4 Interconnect
I2C0[(3)]	0x44E0_B000	0x44E0_BFFF	4KB	Module
	0x44E0_C000	0x44E0_CFFF	4KB	L4 Interconnect
ADC_TSC[(3)]	0x44E0_D000	0x44E0_EFFF	8KB	Module
	0x44E0_F000	0x44E0_FFFF	4KB	L4 Interconnect
Control Module[(3)]	0x44E1_0000	0x44E2_FFFF	128KB	Module
	0x44E3_0000	0x44E3_0FFF	4KB	L4 Interconnect
DMTIMER1_1MS[(3)] (Accurate 1ms timer)	0x44E3_1000	0x44E3_1FFF	4KB	Module
	0x44E3_2000	0x44E3_2FFF	4KB	L4 Interconnect
WDT0[(3)]	0x44E3_3000	0x44E3_3FFF	4KB	Module
	0x44E3_4000	0x44E3_4FFF	4KB	L4 Interconnect
WDT1[(3)]	0x44E3_5000	0x44E3_5FFF	4KB	Module
	0x44E3_6000	0x44E3_6FFF	4KB	L4 Interconnect
SmartReflex0[(3)]	0x44E3_7000	0x44E3_7FFF	4KB	Module

(1) These addresses correspond to L3 region for which the Wakeup M3 has no connectivity. Access to these regions will result in a fault.

(2) Some memory locations map to M3 bit-band alias regions – access to these locations will result in an access to the corresponding bit-band region which maps to an L3 region inaccessible to the Wakeup. Accesses to these regions will result in a fault.

(3) Peripherals located in these regions are accessible by L3 Initiators.

(4) The Wakeup M3 may access the I/D MEM and SMEM using the L4_WKUP (through system bus AHB2OCP interface), or it may use its direct buses (ICode, Dcode), by accessing address 0x00000000 – 0x004FFFFF.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 3-6. L4_WKUP Peripheral Memory Map (continued)

Region Name	Start Address (hex)	End Address (hex)	Size	Description
	0x44E3_8000	0x44E3_8FFF	4KB	L4 Interconnect
SmartReflex1[(3)]	0x44E3_9000	0x44E3_9FFF	4KB	Module
	0x44E3_A000	0x44E3_AFFF	4KB	L4 Interconnect
Reserved[(3)]	0x44E3_B000	0x44E3_DFFF	12KB	Reserved
RTCSS[(5)] [(3)]	0x44E3_E000	0x44E3_EFFF	4KB	Module
	0x44E3_F000	0x44E3_FFFF	4KB	L4 Interconnect
DebugSS Instrumentation HWMaster1 Port[(3)]	0x44E4_0000	0x44E7_FFFF	256KB	Module
	0x44E8_0000	0x44E8_0FFF	4KB	L4 Interconnect
Reserved[(3)]	0x44E8_1000	0x44EF_FFFF	508KB	Reserved
Reserved[(3)]	0x44F0_0000	0x44FF_FFFF	1MB	Reserved
Reserved L3 Slave Region[(1)] [(2)]	0x4500_0000	0xFFFF_FFFF	3G-80MB	Reserved

(5) RTC Subsystem includes RTC module and 32KHz oscillator.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Chapter 4

SPRUH73–July 2011

Programmable Real-Time Unit Subsystem (PRUSS)

This chapter describes the PRUSS for the device.

Topic **Page**

4.1	Introduction	338
4.2	Integration	339
4.3	Functional Description	339
4.4	Use Cases	339
4.5	PRUSS_INTC Registers	339
4.6	PRUSS_PRU_CTRL Registers	404
4.7	PRUSS_PRU_DEBUG Registers	413
4.8	PRUSS_CFG Registers	471
4.9	PRUSS_UART Registers	486
4.10	PRUSS_IEP Registers	500
4.11	PRUSS_ECAP Registers	550
4.12	PRUSS_MII_RT_CFG Registers	565
4.13	PRUSS_MII_MDIO Registers	581

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.1 Introduction

The Programmable Real-time Unit Subsystem (PRUSS) consists of dual 32-bit RISC cores (Programmable Real-Time Units, or PRUs), shared, data, and instruction memories, internal peripheral modules, and an interrupt controller (INTC). The programmable nature of the PRUs, along with their access to pins and events, provide flexibility in implementing custom peripheral interfaces, fast real-time responses, power saving techniques, specialized data handling and DMA operations, and in offloading tasks from the other processor cores of the system-on-chip (SoC).

Figure 4-1 shows the PRUSS details. The subsystem available on this device is the next-generation PRUSS (PRUSSv2). Compared to the previous generation available on AM1x and OMAP-L13x, this version includes the following enhancements:

- Additional data memory (7.5 KB) and instruction memory (4KB)
- 12 KB Shared RAM
- Enhanced GPIO (EGPIO), adding serial, parallel, and MII capture of the PRU input/output pins
- Scratch pad (SPAD) shared by the PRU cores
- Multiplier with optional accumulation (MAC)
- Internal peripheral modules (UART, eCAP, MII_RT, MDIO, and IEP)

Similar to the previous generation, the PRUs have access to all resources on the SoC through the Master OCP port, and the external Host processors can access the PRUSS resources through the Slave OCP port. The Switched Central Resource (SCR) connects the various internal and external masters to the resources inside the PRUSS. The INTC handles system input events and posts events back to the device-level host CPU.

The PRU cores are programmed with a small, deterministic instruction set. Each PRU can operate independently or in coordination with each other and can also work in coordination with the device-level host CPU. This interaction between processors is determined by the nature of the firmware loaded into the PRU's instruction memory.

Figure 4-1. Block Diagram

4.1.1 Features

The PRU subsystem includes the following main features:

- Two PRUs each with:
  - 8KB program memory
  - 8KB data memory

Copyright © 2011, Texas Instruments Incorporated

  - High Performance OCP Master port for accessing external memories
  - Enhanced GPIO with async capture and serial support
  - Multiplier with optional Accumulation
- One Scratch Pad memory and broadside direct connect
  - 3 Banks of 30 32-bit registers
- 12 KB general purpose shared memory
- One Interrupt Controller (INTC)
  - Up to 64 input events supported
  - Interrupt mapping to 10 interrupt channels
  - 10 Host interrupts (2 to PRU0 and PRU1, 8 output to chip level)
  - Each system event can be enabled and disabled
  - Each host event can be enabled and disabled
  - Hardware prioritization of events
- 16 software Events generation by 2 PRUs
- One Ethernet MII_RT module with 2 MII ports and configurable connections to PRUs
- One MDIO Port
- One Industrial Ethernet Peripheral (IEP) to manage/generate Industrial Ethernet functions
  - 1 Industrial Ethernet timer with 7/9 capture and 8 compare events
  - 2 Industrial Ethernet sync signals
  - 2 Industrial Ethernet 16-bit watchdog timers
  - Industrial Ethernet digital IOs
- One 16550-compatible UART with a dedicated 192 MHz clock
- One Enhanced Capture Module (ECAP)
- Flexible power management support
- Integrated switched central resource (SCR) bus for connecting the various internal and external masters to the resources inside the PRUSS
- OCP Slave port for external masters to access PRUSS memories
- Optional address translation for PRU transaction to External Host
- All memories within the PRUSS support parity

4.2 Integration

4.3 Functional Description

4.4 Use Cases

4.5 PRUSS_INTC Registers

Table 4-1 lists the memory-mapped registers for the PRUSS_INTC. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 4-1 should be considered as reserved locations and the register contents should not be modified.

Table 4-1. PRUSS_INTC REGISTERS

Offset	Acronym	Register Description	Section
0h	REVID	REVISION REGISTER	Section 14.4.3.25
400h	CMR0	INTERRUPT CHANNEL MAP REGISTER FOR 0 TO 0+3	Section 4.5.2

Copyright © 2011, Texas Instruments Incorporated

Table 4-1. PRUSS_INTC REGISTERS (continued)

Offset	Acronym	Register Description	Section
404h	CMR1	INTERRUPT CHANNEL MAP REGISTER FOR 4 TO 4+3	Section 4.5.3
408h	CMR2	INTERRUPT CHANNEL MAP REGISTER FOR 8 TO 8+3	Section 4.5.4
40Ch	CMR3	INTERRUPT CHANNEL MAP REGISTER FOR 12 TO 12+3	Section 4.5.5
410h	CMR4	INTERRUPT CHANNEL MAP REGISTER FOR 16 TO 16+3	Section 4.5.6
414h	CMR5	INTERRUPT CHANNEL MAP REGISTER FOR 20 TO 20+3	Section 4.5.7
418h	CMR6	INTERRUPT CHANNEL MAP REGISTER FOR 24 TO 24+3	Section 4.5.8
41Ch	CMR7	INTERRUPT CHANNEL MAP REGISTER FOR 28 TO 28+3	Section 4.5.9
420h	CMR8	INTERRUPT CHANNEL MAP REGISTER FOR 32 TO 32+3	Section 4.5.10
424h	CMR9	INTERRUPT CHANNEL MAP REGISTER FOR 36 TO 36+3	Section 4.5.11
428h	CMR10	INTERRUPT CHANNEL MAP REGISTER FOR 40 TO 40+3	Section 4.5.12
42Ch	CMR11	INTERRUPT CHANNEL MAP REGISTER FOR 44 TO 44+3	Section 4.5.13
430h	CMR12	INTERRUPT CHANNEL MAP REGISTER FOR 48 TO 48+3	Section 4.5.14
434h	CMR13	INTERRUPT CHANNEL MAP REGISTER FOR 52 TO 52+3	Section 4.5.15
438h	CMR14	INTERRUPT CHANNEL MAP REGISTER FOR 56 TO 56+3	Section 4.5.16
43Ch	CMR15	INTERRUPT CHANNEL MAP REGISTER FOR 60 TO 60+3	Section 4.5.17
80h	GPIR	GLOBAL PRIORITIZED INTERRUPT REGISTER	Section 4.5.18
10h	GER	GLOBAL HOST INT ENABLE REGISTER	Section 4.5.19
800h	HMR0	HOST INTERRUPT MAP REGISTER FOR 0 TO 0+3	Section 4.5.20
804h	HMR1	HOST INTERRUPT MAP REGISTER FOR 4 TO 4+3	Section 4.5.21
808h	HMR2	HOST INTERRUPT MAP REGISTER FOR 8 TO 8+3	Section 4.5.22
900h	HIPIR0	HOST INT 0 PRIORITIZED INTERRUPT REGISTER	Section 4.5.23
904h	HIPIR1	HOST INT 1 PRIORITIZED INTERRUPT REGISTER	Section 4.5.24
908h	HIPIR2	HOST INT 2 PRIORITIZED INTERRUPT REGISTER	Section 4.5.25
90Ch	HIPIR3	HOST INT 3 PRIORITIZED INTERRUPT REGISTER	Section 4.5.26
910h	HIPIR4	HOST INT 4 PRIORITIZED INTERRUPT REGISTER	Section 4.5.27
914h	HIPIR5	HOST INT 5 PRIORITIZED INTERRUPT REGISTER	Section 4.5.28
918h	HIPIR6	HOST INT 6 PRIORITIZED INTERRUPT REGISTER	Section 4.5.29
91Ch	HIPIR7	HOST INT 7 PRIORITIZED INTERRUPT REGISTER	Section 4.5.30
920h	HIPIR8	HOST INT 8 PRIORITIZED INTERRUPT REGISTER	Section 4.5.31
924h	HIPIR9	HOST INT 9 PRIORITIZED INTERRUPT REGISTER	Section 4.5.32
1Ch	GNLR	GLOBAL NESTING LEVEL REGISTER	Section 4.5.33
20h	SISR	STATUS SET INDEX REGISTER	Section 4.5.34
D00h	SIPR0	POLARITY REGISTER 0	Section 4.5.35
D04h	SIPR1	POLARITY REGISTER 1	Section 4.5.36
D80h	SITR0	TYPE REGISTER 0	Section 4.5.37
D84h	SITR1	TYPE REGISTER 1	Section 4.5.38
24h	SICR	STATUS CLEAR INDEX REGISTER	Section 4.5.39

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 4-1. PRUSS_INTC REGISTERS (continued)

Offset	Acronym	Register Description	Section
4h	CR	CONTROL REGISTER	Section 4.5.40
28h	EISR	ENABLE SET INDEX REGISTER	Section 4.5.41
1100h	HINLR0	HOST INT 0 NESTING LEVEL REGISTER	Section 4.5.42
1104h	HINLR1	HOST INT 1 NESTING LEVEL REGISTER	Section 4.5.43
1108h	HINLR2	HOST INT 2 NESTING LEVEL REGISTER	Section 4.5.44
110Ch	HINLR3	HOST INT 3 NESTING LEVEL REGISTER	Section 4.5.45
1110h	HINLR4	HOST INT 4 NESTING LEVEL REGISTER	Section 4.5.46
1114h	HINLR5	HOST INT 5 NESTING LEVEL REGISTER	Section 4.5.47
1118h	HINLR6	HOST INT 6 NESTING LEVEL REGISTER	Section 4.5.48
111Ch	HINLR7	HOST INT 7 NESTING LEVEL REGISTER	Section 4.5.49
1120h	HINLR8	HOST INT 8 NESTING LEVEL REGISTER	Section 4.5.50
1124h	HINLR9	HOST INT 9 NESTING LEVEL REGISTER	Section 4.5.51
2Ch	EICR	ENABLE CLEAR INDEX REGISTER	Section 4.5.52
200h	SRSR0	RAW STATUS REGISTER 0	Section 4.5.53
204h	SRSR1	RAW STATUS REGISTER 1	Section 4.5.54
34h	HIEISR	HOST INT ENABLE SET INDEX REGISTER	Section 4.5.55
1500h	HIER	HOST INT ENABLE REGISTER 0	Section 4.5.56
38h	HIDISR	HOST INT ENABLE CLEAR INDEX REGISTER	Section 4.5.57
280h	SECR0	ENABLED STATUS REGISTER 0	Section 4.5.58
284h	SECR1	ENABLED STATUS REGISTER 1	Section 4.5.59
300h	ESR0	ENABLE SET REGISTER 0	Section 4.5.60
304h	ERS1	ENABLE SET REGISTER 1	Section 4.5.61
380h	ECR0	ENABLE CLEAR REGISTER 0	Section 4.5.62
384h	ECR1	ENABLE CLEAR REGISTER 1	Section 4.5.63

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.1 REVID Register (offset = 0h) [reset = 4E82A900h]

REVISION REGISTER

REVID is shown in Figure 4-131 and described in Table 4-134.

Figure 4-2. REVID Register

31	30	29	28	27	26	25	24
REV_SCHEME		Reserved		REV_MODULE			
R-1h				R-E82h			

23	22	21	20	19	18	17	16
REV_MODULE							
R-E82h							

15	14	13	12	11	10	9	8
REV_RTL					REV_MAJOR		
R-15h					R-1h		

7	6	5	4	3	2	1	0
REV_CUSTOM		REV_MINOR					
R-0h		R-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-2. REVID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	REV_SCHEME	R	1h	SCHEME
27-16	REV_MODULE	R	E82h	MODULE ID
15-11	REV_RTL	R	15h	RTL REVISIONS
10-8	REV_MAJOR	R	1h	MAJOR REVISION
7-6	REV_CUSTOM	R	0h	CUSTOM REVISION
5-0	REV_MINOR	R	0h	MINOR REVISION

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.2 CMR0 Register (offset = 400h) [reset = 0h]

INTERRUPT CHANNEL MAP REGISTER FOR 0 TO 0+3

CMR0 is shown in Figure 4-3 and described in Table 4-3.

Figure 4-3. CMR0 Register

31	30	29	28	27	26	25	24
Reserved				CH_MAP_3			
				R/W-0h			
23	22	21	20	19	18	17	16
Reserved				CH_MAP_2			
				R/W-0h			
15	14	13	12	11	10	9	8
Reserved				CH_MAP_1			
				R/W-0h			
7	6	5	4	3	2	1	0
Reserved				CH_MAP_0			
				R/W-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-3. CMR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
27-24	CH_MAP_3	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[3]
19-16	CH_MAP_2	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[2]
11-8	CH_MAP_1	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[1]
3-0	CH_MAP_0	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[0]

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.3 CMR1 Register (offset = 404h) [reset = 0h]

INTERRUPT CHANNEL MAP REGISTER FOR 4 TO 4+3

CMR1 is shown in Figure 4-4 and described in Table 4-4.

Figure 4-4. CMR1 Register

31	30	29	28	27	26	25	24
Reserved				CH_MAP_7			
				R/W-0h			
23	22	21	20	19	18	17	16
Reserved				CH_MAP_6			
				R/W-0h			
15	14	13	12	11	10	9	8
Reserved				CH_MAP_5			
				R/W-0h			
7	6	5	4	3	2	1	0
Reserved				CH_MAP_4			
				R/W-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-4. CMR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
27-24	CH_MAP_7	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[7]
19-16	CH_MAP_6	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[6]
11-8	CH_MAP_5	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[5]
3-0	CH_MAP_4	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[4]

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.4 CMR2 Register (offset = 408h) [reset = 0h]

INTERRUPT CHANNEL MAP REGISTER FOR 8 TO 8+3

CMR2 is shown in Figure 4-5 and described in Table 4-5.

Figure 4-5. CMR2 Register

31	30	29	28	27	26	25	24
Reserved				CH_MAP_11			
				R/W-0h			

23	22	21	20	19	18	17	16
Reserved				CH_MAP_10			
				R/W-0h			

15	14	13	12	11	10	9	8
Reserved				CH_MAP_9			
				R/W-0h			

7	6	5	4	3	2	1	0
Reserved				CH_MAP_8			
				R/W-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-5. CMR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
27-24	CH_MAP_11	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[11]
19-16	CH_MAP_10	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[10]
11-8	CH_MAP_9	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[9]
3-0	CH_MAP_8	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[8]

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.5 CMR3 Register (offset = 40Ch) [reset = 0h]

INTERRUPT CHANNEL MAP REGISTER FOR 12 TO 12+3

CMR3 is shown in Figure 4-6 and described in Table 4-6.

Figure 4-6. CMR3 Register

31	30	29	28	27	26	25	24
Reserved				CH_MAP_15			
				R/W-0h			
23	22	21	20	19	18	17	16
Reserved				CH_MAP_14			
				R/W-0h			
15	14	13	12	11	10	9	8
Reserved				CH_MAP_13			
				R/W-0h			
7	6	5	4	3	2	1	0
Reserved				CH_MAP_12			
				R/W-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-6. CMR3 Register Field Descriptions

Bit	Field	Type	Reset	Description
27-24	CH_MAP_15	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[15]
19-16	CH_MAP_14	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[14]
11-8	CH_MAP_13	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[13]
3-0	CH_MAP_12	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[12]

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.6 CMR4 Register (offset = 410h) [reset = 0h]

INTERRUPT CHANNEL MAP REGISTER FOR 16 TO 16+3

CMR4 is shown in Figure 4-7 and described in Table 4-7.

Figure 4-7. CMR4 Register

31	30	29	28	27	26	25	24
Reserved				CH_MAP_19			
				R/W-0h			

23	22	21	20	19	18	17	16
Reserved				CH_MAP_18			
				R/W-0h			

15	14	13	12	11	10	9	8
Reserved				CH_MAP_17			
				R/W-0h			

7	6	5	4	3	2	1	0
Reserved				CH_MAP_16			
				R/W-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-7. CMR4 Register Field Descriptions

Bit	Field	Type	Reset	Description
27-24	CH_MAP_19	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[19]
19-16	CH_MAP_18	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[18]
11-8	CH_MAP_17	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[17]
3-0	CH_MAP_16	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[16]

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.7 CMR5 Register (offset = 414h) [reset = 0h]

INTERRUPT CHANNEL MAP REGISTER FOR 20 TO 20+3

CMR5 is shown in Figure 4-8 and described in Table 4-8.

Figure 4-8. CMR5 Register

31	30	29	28	27	26	25	24
Reserved				CH_MAP_23			
				R/W-0h			

23	22	21	20	19	18	17	16
Reserved				CH_MAP_22			
				R/W-0h			

15	14	13	12	11	10	9	8
Reserved				CH_MAP_21			
				R/W-0h			

7	6	5	4	3	2	1	0
Reserved				CH_MAP_20			
				R/W-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-8. CMR5 Register Field Descriptions

Bit	Field	Type	Reset	Description
27-24	CH_MAP_23	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[23]
19-16	CH_MAP_22	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[22]
11-8	CH_MAP_21	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[21]
3-0	CH_MAP_20	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[20]

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.8 CMR6 Register (offset = 418h) [reset = 0h]

INTERRUPT CHANNEL MAP REGISTER FOR 24 TO 24+3

CMR6 is shown in Figure 4-9 and described in Table 4-9.

Figure 4-9. CMR6 Register

31	30	29	28	27	26	25	24
Reserved				CH_MAP_27			
				R/W-0h			

23	22	21	20	19	18	17	16
Reserved				CH_MAP_26			
				R/W-0h			

15	14	13	12	11	10	9	8
Reserved				CH_MAP_25			
				R/W-0h			

7	6	5	4	3	2	1	0
Reserved				CH_MAP_24			
				R/W-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-9. CMR6 Register Field Descriptions

Bit	Field	Type	Reset	Description
27-24	CH_MAP_27	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[27]
19-16	CH_MAP_26	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[26]
11-8	CH_MAP_25	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[25]
3-0	CH_MAP_24	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[24]

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.9 CMR7 Register (offset = 41Ch) [reset = 0h]

INTERRUPT CHANNEL MAP REGISTER FOR 28 TO 28+3

CMR7 is shown in Figure 4-10 and described in Table 4-10.

Figure 4-10. CMR7 Register

31	30	29	28	27	26	25	24
Reserved				CH_MAP_31			
				R/W-0h			

23	22	21	20	19	18	17	16
Reserved				CH_MAP_30			
				R/W-0h			

15	14	13	12	11	10	9	8
Reserved				CH_MAP_29			
				R/W-0h			

7	6	5	4	3	2	1	0
Reserved				CH_MAP_28			
				R/W-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-10. CMR7 Register Field Descriptions

Bit	Field	Type	Reset	Description
27-24	CH_MAP_31	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[31]
19-16	CH_MAP_30	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[30]
11-8	CH_MAP_29	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[29]
3-0	CH_MAP_28	R/W	0h	INTERRUPT CHANNEL MAP FOR INTR_IN[28]

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.10 CMR8 Register (offset = 420h) [reset = 0h]

INTERRUPT CHANNEL MAP REGISTER FOR 32 TO 32+3

CMR8 is shown in Figure 4-11 and described in Table 4-11.

Figure 4-11. CMR8 Register

31	30	29	28	27	26	25	24
Reserved				CH_MAP_35			
				R/W-0h			

23	22	21	20	19	18	17	16
Reserved				CH_MAP_34			
				R/W-0h			

15	14	13	12	11	10	9	8
Reserved				CH_MAP_33			
				R/W-0h			

7	6	5	4	3	2	1	0
Reserved				CH_MAP_32			
				R/W-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-11. CMR8 Register Field Descriptions

Bit	Field	Type	Reset	Description
27-24	CH_MAP_35	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[3]
19-16	CH_MAP_34	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[2]
11-8	CH_MAP_33	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[1]
3-0	CH_MAP_32	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[0]

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.11 CMR9 Register (offset = 424h) [reset = 0h]

INTERRUPT CHANNEL MAP REGISTER FOR 36 TO 36+3

CMR9 is shown in Figure 4-12 and described in Table 4-12.

Figure 4-12. CMR9 Register

31	30	29	28	27	26	25	24
Reserved				CH_MAP_39			
				R/W-0h			

23	22	21	20	19	18	17	16
Reserved				CH_MAP_38			
				R/W-0h			

15	14	13	12	11	10	9	8
Reserved				CH_MAP_37			
				R/W-0h			

7	6	5	4	3	2	1	0
Reserved				CH_MAP_36			
				R/W-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-12. CMR9 Register Field Descriptions

Bit	Field	Type	Reset	Description
27-24	CH_MAP_39	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[7]
19-16	CH_MAP_38	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[6]
11-8	CH_MAP_37	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[5]
3-0	CH_MAP_36	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[4]

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.12 CMR10 Register (offset = 428h) [reset = 0h]

INTERRUPT CHANNEL MAP REGISTER FOR 40 TO 40+3

CMR10 is shown in Figure 4-13 and described in Table 4-13.

Figure 4-13. CMR10 Register

31	30	29	28	27	26	25	24
Reserved				CH_MAP_43			
				R/W-0h			
23	22	21	20	19	18	17	16
Reserved				CH_MAP_42			
				R/W-0h			
15	14	13	12	11	10	9	8
Reserved				CH_MAP_41			
				R/W-0h			
7	6	5	4	3	2	1	0
Reserved				CH_MAP_40			
				R/W-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-13. CMR10 Register Field Descriptions

Bit	Field	Type	Reset	Description
27-24	CH_MAP_43	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[11]
19-16	CH_MAP_42	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[10]
11-8	CH_MAP_41	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[9]
3-0	CH_MAP_40	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[8]

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.13 CMR11 Register (offset = 42Ch) [reset = 0h]

INTERRUPT CHANNEL MAP REGISTER FOR 44 TO 44+3

CMR11 is shown in Figure 4-14 and described in Table 4-14.

Figure 4-14. CMR11 Register

31	30	29	28	27	26	25	24
Reserved				CH_MAP_47			
				R/W-0h			

23	22	21	20	19	18	17	16
Reserved				CH_MAP_46			
				R/W-0h			

15	14	13	12	11	10	9	8
Reserved				CH_MAP_45			
				R/W-0h			

7	6	5	4	3	2	1	0
Reserved				CH_MAP_44			
				R/W-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-14. CMR11 Register Field Descriptions

Bit	Field	Type	Reset	Description
27-24	CH_MAP_47	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[15]
19-16	CH_MAP_46	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[14]
11-8	CH_MAP_45	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[13]
3-0	CH_MAP_44	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[12]

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.14 CMR12 Register (offset = 430h) [reset = 0h]

INTERRUPT CHANNEL MAP REGISTER FOR 48 TO 48+3

CMR12 is shown in Figure 4-15 and described in Table 4-15.

Figure 4-15. CMR12 Register

31	30	29	28	27	26	25	24
Reserved				CH_MAP_51			
				R/W-0h			

23	22	21	20	19	18	17	16
Reserved				CH_MAP_50			
				R/W-0h			

15	14	13	12	11	10	9	8
Reserved				CH_MAP_49			
				R/W-0h			

7	6	5	4	3	2	1	0
Reserved				CH_MAP_48			
				R/W-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-15. CMR12 Register Field Descriptions

Bit	Field	Type	Reset	Description
27-24	CH_MAP_51	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[19]
19-16	CH_MAP_50	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[18]
11-8	CH_MAP_49	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[17]
3-0	CH_MAP_48	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[16]

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.15 CMR13 Register (offset = 434h) [reset = 0h]

INTERRUPT CHANNEL MAP REGISTER FOR 52 TO 52+3

CMR13 is shown in Figure 4-16 and described in Table 4-16.

Figure 4-16. CMR13 Register

31	30	29	28	27	26	25	24
Reserved				CH_MAP_55			
				R/W-0h			
23	22	21	20	19	18	17	16
Reserved				CH_MAP_54			
				R/W-0h			
15	14	13	12	11	10	9	8
Reserved				CH_MAP_53			
				R/W-0h			
7	6	5	4	3	2	1	0
Reserved				CH_MAP_52			
				R/W-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-16. CMR13 Register Field Descriptions

Bit	Field	Type	Reset	Description
27-24	CH_MAP_55	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[23]
19-16	CH_MAP_54	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[22]
11-8	CH_MAP_53	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[21]
3-0	CH_MAP_52	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[20]

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.16 CMR14 Register (offset = 438h) [reset = 0h]

INTERRUPT CHANNEL MAP REGISTER FOR 56 TO 56+3

CMR14 is shown in Figure 4-17 and described in Table 4-17.

Figure 4-17. CMR14 Register

31	30	29	28	27	26	25	24
Reserved				CH_MAP_59			
				R/W-0h			

23	22	21	20	19	18	17	16
Reserved				CH_MAP_58			
				R/W-0h			

15	14	13	12	11	10	9	8
Reserved				CH_MAP_57			
				R/W-0h			

7	6	5	4	3	2	1	0
Reserved				CH_MAP_56			
				R/W-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-17. CMR14 Register Field Descriptions

Bit	Field	Type	Reset	Description
27-24	CH_MAP_59	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[27]
19-16	CH_MAP_58	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[26]
11-8	CH_MAP_57	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[25]
3-0	CH_MAP_56	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[24]

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.17 CMR15 Register (offset = 43Ch) [reset = 0h]

INTERRUPT CHANNEL MAP REGISTER FOR 60 TO 60+3

CMR15 is shown in Figure 4-18 and described in Table 4-18.

Figure 4-18. CMR15 Register

31	30	29	28	27	26	25	24
Reserved				CH_MAP_63			
				R/W-0h			

23	22	21	20	19	18	17	16
Reserved				CH_MAP_62			
				R/W-0h			

15	14	13	12	11	10	9	8
Reserved				CH_MAP_61			
				R/W-0h			

7	6	5	4	3	2	1	0
Reserved				CH_MAP_60			
				R/W-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-18. CMR15 Register Field Descriptions

Bit	Field	Type	Reset	Description
27-24	CH_MAP_63	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[31]
19-16	CH_MAP_62	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[30]
11-8	CH_MAP_61	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[29]
3-0	CH_MAP_60	R/W	0h	INTERRUPT CHANNEL MAP FOR SLV_EVENTS_IN[28]

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.18 GPIR Register (offset = 80h) [reset = 80000000h]

GLOBAL PRIORITIZED INTERRUPT REGISTER

GPIR is shown in Figure 4-19 and described in Table 4-19.

Figure 4-19. GPIR Register

31	30	29	28	27	26	25	24
GLB_NONE	Reserved						
R-1h							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved						GLB_PRI_INTR	
						R-0h	

7	6	5	4	3	2	1	0
GLB_PRI_INTR							
R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-19. GPIR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GLB_NONE	R	1h	NO INTERRUPT PENDING FLAG
9-0	GLB_PRI_INTR	R	0h	PRIORITIZED INTERRUPT

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.19 GER Register (offset = 10h) [reset = 0h]

GLOBAL HOST INT ENABLE REGISTER

GER is shown in Figure 4-20 and described in Table 4-20.

Figure 4-20. GER Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															ENABLE_HINT_ANY
																															R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-20. GER Register Field Descriptions

Bit	Field	Type	Reset	Description
0	ENABLE_HINT_ANY	R/W	0h	GLOBAL ENABLE FOR ALL HOST INTS

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.20 HMR0 Register (offset = 800h) [reset = 0h]

HOST INTERRUPT MAP REGISTER FOR 0 TO 0+3

HMR0 is shown in Figure 4-21 and described in Table 4-21.

Figure 4-21. HMR0 Register

31	30	29	28	27	26	25	24
Reserved				HINT_MAP_3			
				R/W-0h			

23	22	21	20	19	18	17	16
Reserved				HINT_MAP_2			
				R/W-0h			

15	14	13	12	11	10	9	8
Reserved				HINT_MAP_1			
				R/W-0h			

7	6	5	4	3	2	1	0
Reserved				HINT_MAP_0			
				R/W-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-21. HMR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
27-24	HINT_MAP_3	R/W	0h	HOST INTERRUPT MAP FOR CHANNEL 3
19-16	HINT_MAP_2	R/W	0h	HOST INTERRUPT MAP FOR CHANNEL 2
11-8	HINT_MAP_1	R/W	0h	HOST INTERRUPT MAP FOR CHANNEL 1
3-0	HINT_MAP_0	R/W	0h	HOST INTERRUPT MAP FOR CHANNEL 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.21 HMR1 Register (offset = 804h) [reset = 0h]

HOST INTERRUPT MAP REGISTER FOR 4 TO 4+3

HMR1 is shown in Figure 4-22 and described in Table 4-22.

Figure 4-22. HMR1 Register

31	30	29	28	27	26	25	24
Reserved				HINT_MAP_7			
				R/W-0h			

23	22	21	20	19	18	17	16
Reserved				HINT_MAP_6			
				R/W-0h			

15	14	13	12	11	10	9	8
Reserved				HINT_MAP_5			
				R/W-0h			

7	6	5	4	3	2	1	0
Reserved				HINT_MAP_4			
				R/W-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-22. HMR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
27-24	HINT_MAP_7	R/W	0h	HOST INTERRUPT MAP FOR CHANNEL 7
19-16	HINT_MAP_6	R/W	0h	HOST INTERRUPT MAP FOR CHANNEL 6
11-8	HINT_MAP_5	R/W	0h	HOST INTERRUPT MAP FOR CHANNEL 5
3-0	HINT_MAP_4	R/W	0h	HOST INTERRUPT MAP FOR CHANNEL 4

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.22 HMR2 Register (offset = 808h) [reset = 0h]

HOST INTERRUPT MAP REGISTER FOR 8 TO 8+3

HMR2 is shown in Figure 4-23 and described in Table 4-23.

Figure 4-23. HMR2 Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved				HINT_MAP_9			
				R/W-0h			

7	6	5	4	3	2	1	0
Reserved				HINT_MAP_8			
				R/W-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-23. HMR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
11-8	HINT_MAP_9	R/W	0h	HOST INTERRUPT MAP FOR CHANNEL 9
3-0	HINT_MAP_8	R/W	0h	HOST INTERRUPT MAP FOR CHANNEL 8

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.23 HIPIR0 Register (offset = 900h) [reset = 80000000h]

HOST INT 0 PRIORITIZED INTERRUPT REGISTER

HIPIR0 is shown in Figure 4-24 and described in Table 4-24.

Figure 4-24. HIPIR0 Register

31	30	29	28	27	26	25	24
NONE_HINT_0	Reserved						
R-1h							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved						PRI_HINT_0	
						R-0h	

7	6	5	4	3	2	1	0
PRI_HINT_0							
R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-24. HIPIR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NONE_HINT_0	R	1h	NO INTERRUPT PENDING FLAG
9-0	PRI_HINT_0	R	0h	HOST INT 0 PRIORITIZED INTERRUPT

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.24 HIPIR1 Register (offset = 904h) [reset = 80000000h]

HOST INT 1 PRIORITIZED INTERRUPT REGISTER

HIPIR1 is shown in Figure 4-25 and described in Table 4-25.

Figure 4-25. HIPIR1 Register

31	30	29	28	27	26	25	24
NONE_HINT_1	Reserved						
R-1h							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved						PRI_HINT_1	
						R-0h	

7	6	5	4	3	2	1	0
PRI_HINT_1							
R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-25. HIPIR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NONE_HINT_1	R	1h	NO INTERRUPT PENDING FLAG
9-0	PRI_HINT_1	R	0h	HOST INT 1 PRIORITIZED INTERRUPT

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.25 HIPIR2 Register (offset = 908h) [reset = 80000000h]

HOST INT 2 PRIORITIZED INTERRUPT REGISTER

HIPIR2 is shown in Figure 4-26 and described in Table 4-26.

Figure 4-26. HIPIR2 Register

31	30	29	28	27	26	25	24
NONE_HINT_2	Reserved						
R-1h							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved						PRI_HINT_2	
						R-0h	

7	6	5	4	3	2	1	0
PRI_HINT_2							
R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-26. HIPIR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NONE_HINT_2	R	1h	NO INTERRUPT PENDING FLAG
9-0	PRI_HINT_2	R	0h	HOST INT 2 PRIORITIZED INTERRUPT

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.26 HIPIR3 Register (offset = 90Ch) [reset = 80000000h]

HOST INT 3 PRIORITIZED INTERRUPT REGISTER

HIPIR3 is shown in Figure 4-27 and described in Table 4-27.

Figure 4-27. HIPIR3 Register

31	30	29	28	27	26	25	24
NONE_HINT_3	Reserved						
R-1h							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved						PRI_HINT_3	
						R-0h	

7	6	5	4	3	2	1	0
PRI_HINT_3							
R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-27. HIPIR3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NONE_HINT_3	R	1h	NO INTERRUPT PENDING FLAG
9-0	PRI_HINT_3	R	0h	HOST INT 3 PRIORITIZED INTERRUPT

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.27 HIPIR4 Register (offset = 910h) [reset = 80000000h]

HOST INT 4 PRIORITIZED INTERRUPT REGISTER

HIPIR4 is shown in Figure 4-28 and described in Table 4-28.

Figure 4-28. HIPIR4 Register

31	30	29	28	27	26	25	24
NONE_HINT_4	Reserved						
R-1h							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved						PRI_HINT_4	
						R-0h	

7	6	5	4	3	2	1	0
PRI_HINT_4							
R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-28. HIPIR4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NONE_HINT_4	R	1h	NO INTERRUPT PENDING FLAG
9-0	PRI_HINT_4	R	0h	HOST INT 4 PRIORITIZED INTERRUPT

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.28 HIPIR5 Register (offset = 914h) [reset = 80000000h]

HOST INT 5 PRIORITIZED INTERRUPT REGISTER

HIPIR5 is shown in Figure 4-29 and described in Table 4-29.

Figure 4-29. HIPIR5 Register

31	30	29	28	27	26	25	24
NONE_HINT_5	Reserved						
R-1h							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved						PRI_HINT_5	
						R-0h	

7	6	5	4	3	2	1	0
PRI_HINT_5							
R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-29. HIPIR5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NONE_HINT_5	R	1h	NO INTERRUPT PENDING FLAG
9-0	PRI_HINT_5	R	0h	HOST INT 5 PRIORITIZED INTERRUPT

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.29 HIPIR6 Register (offset = 918h) [reset = 80000000h]

HOST INT 6 PRIORITIZED INTERRUPT REGISTER

HIPIR6 is shown in Figure 4-30 and described in Table 4-30.

Figure 4-30. HIPIR6 Register

31	30	29	28	27	26	25	24
NONE_HINT_6	Reserved						
R-1h							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved						PRI_HINT_6	
						R-0h	

7	6	5	4	3	2	1	0
PRI_HINT_6							
R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-30. HIPIR6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NONE_HINT_6	R	1h	NO INTERRUPT PENDING FLAG
9-0	PRI_HINT_6	R	0h	HOST INT 6 PRIORITIZED INTERRUPT

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.30 HIPIR7 Register (offset = 91Ch) [reset = 80000000h]

HOST INT 7 PRIORITIZED INTERRUPT REGISTER

HIPIR7 is shown in Figure 4-31 and described in Table 4-31.

Figure 4-31. HIPIR7 Register

31	30	29	28	27	26	25	24
NONE_HINT_7	Reserved						
R-1h							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved						PRI_HINT_7	
						R-0h	

7	6	5	4	3	2	1	0
PRI_HINT_7							
R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-31. HIPIR7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NONE_HINT_7	R	1h	NO INTERRUPT PENDING FLAG
9-0	PRI_HINT_7	R	0h	HOST INT 7 PRIORITIZED INTERRUPT

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.31 HIPIR8 Register (offset = 920h) [reset = 80000000h]

HOST INT 8 PRIORITIZED INTERRUPT REGISTER

HIPIR8 is shown in Figure 4-32 and described in Table 4-32.

Figure 4-32. HIPIR8 Register

31	30	29	28	27	26	25	24
NONE_HINT_8	Reserved						
R-1h							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved						PRI_HINT_8	
						R-0h	

7	6	5	4	3	2	1	0
PRI_HINT_8							
R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-32. HIPIR8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NONE_HINT_8	R	1h	NO INTERRUPT PENDING FLAG
9-0	PRI_HINT_8	R	0h	HOST INT 8 PRIORITIZED INTERRUPT

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.32 HIPIR9 Register (offset = 924h) [reset = 80000000h]

HOST INT 9 PRIORITIZED INTERRUPT REGISTER

HIPIR9 is shown in Figure 4-33 and described in Table 4-33.

Figure 4-33. HIPIR9 Register

31	30	29	28	27	26	25	24
NONE_HINT_9	Reserved						
R-1h							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved						PRI_HINT_9	
						R-0h	

7	6	5	4	3	2	1	0
PRI_HINT_9							
R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-33. HIPIR9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NONE_HINT_9	R	1h	NO INTERRUPT PENDING FLAG
9-0	PRI_HINT_9	R	0h	HOST INT 9 PRIORITIZED INTERRUPT

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.33 GNLR Register (offset = 1Ch) [reset = 100h]

GLOBAL NESTING LEVEL REGISTER

GNLR is shown in Figure 4-34 and described in Table 4-34.

Figure 4-34. GNLR Register

31	30	29	28	27	26	25	24
AUTO_OVERRIDE	Reserved						
W-0h							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							GLB_NEST_LEVEL
							R/W-100h

7	6	5	4	3	2	1	0
GLB_NEST_LEVEL							
R/W-100h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-34. GNLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	AUTO_OVERRIDE	W	0h	AUTO NESTING
8-0	GLB_NEST_LEVEL	R/W	100h	GLOBAL NESTING LEVEL

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.34 SISR Register (offset = 20h) [reset = 0h]

STATUS SET INDEX REGISTER

SISR is shown in Figure 4-35 and described in Table 4-35.

Figure 4-35. SISR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																						STATUS_SET_INDEX									
																						W-0h									

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-35. SISR Register Field Descriptions

Bit	Field	Type	Reset	Description
9-0	STATUS_SET_INDEX	W	0h	STATUS SET INDEX REGISTER (WRITE INDEX TO SET STATUS OF)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.35 SIPR0 Register (offset = D00h) [reset = 1h]

POLARITY REGISTER 0

SIPR0 is shown in Figure 4-36 and described in Table 4-36.

Figure 4-36. SIPR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POLARITY_31_0																															
R/W-1h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-36. SIPR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	POLARITY_31_0	R/W	1h	POLARITY FOR INTR_IN 0=LOW

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.36 SIPR1 Register (offset = D04h) [reset = 1h]

POLARITY REGISTER 1

SIPR1 is shown in Figure 4-37 and described in Table 4-37.

Figure 4-37. SIPR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POLARITY_63_32																															
R/W-1h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-37. SIPR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	POLARITY_63_32	R/W	1h	POLARITY FOR SLV_EVENTS_IN[0] 0=LOW

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.37 SITR0 Register (offset = D80h) [reset = 0h]

TYPE REGISTER 0

SITR0 is shown in Figure 4-38 and described in Table 4-38.

Figure 4-38. SITR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE_31_0																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-38. SITR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TYPE_31_0	R/W	0h	TYPE FOR INTR_IN[0] 0=LEVEL

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.38 SITR1 Register (offset = D84h) [reset = 0h]

TYPE REGISTER 1

SITR1 is shown in Figure 4-39 and described in Table 4-39.

Figure 4-39. SITR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE_63_32																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-39. SITR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TYPE_63_32	R/W	0h	TYPE FOR SLV_EVENTS_IN[0] 0=LEVEL

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.39 SICR Register (offset = 24h) [reset = 0h]

STATUS CLEAR INDEX REGISTER

SICR is shown in Figure 4-40 and described in Table 4-40.

Figure 4-40. SICR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																						STATUS_CLR_INDEX									
																						W-0h									

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-40. SICR Register Field Descriptions

Bit	Field	Type	Reset	Description
9-0	STATUS_CLR_INDEX	W	0h	STATUS CLEAR INDEX REGISTER (WRITE INDEX TO CLEAR STATUS OF)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.40 CR Register (offset = 4h) [reset = 0h]

CONTROL REGISTER

CR is shown in Figure 4-41 and described in Table 4-41.

Figure 4-41. CR Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved			PRIORITY_HOLD_MODE	NEST_MODE		WAKEUP_MODE	Reserved
			R/W-0h	R/W-0h		R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-41. CR Register Field Descriptions

Bit	Field	Type	Reset	Description
4	PRIORITY_HOLD_MODE	R/W	0h	PRIORITY HOLDING MODE
3-2	NEST_MODE	R/W	0h	NESTING MODE
1	WAKEUP_MODE	R/W	0h	WAKEUP MODE ENABLE

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.41 EISR Register (offset = 28h) [reset = 0h]

ENABLE SET INDEX REGISTER

EISR is shown in Figure 4-42 and described in Table 4-42.

Figure 4-42. EISR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																						ENABLE_SET_INDEX									
																						W-0h									

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-42. EISR Register Field Descriptions

Bit	Field	Type	Reset	Description
9-0	ENABLE_SET_INDEX	W	0h	ENABLE SET INDEX REGISTER (WRITE INDEX TO SET ENABLE OF)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.42 HINLR0 Register (offset = 1100h) [reset = 100h]

HOST INT 0 NESTING LEVEL REGISTER

HINLR0 is shown in Figure 4-43 and described in Table 4-43.

Figure 4-43. HINLR0 Register

31	30	29	28	27	26	25	24
AUTO_OVERRIDE	Reserved						
W-0h							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							NEST_HINT_0
							R/W-100h

7	6	5	4	3	2	1	0
NEST_HINT_0							
R/W-100h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-43. HINLR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	AUTO_OVERRIDE	W	0h	AUTO NESTING
8-0	NEST_HINT_0	R/W	100h	HOST INT 0 NESTING LEVEL

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.43 HINLR1 Register (offset = 1104h) [reset = 100h]

HOST INT 1 NESTING LEVEL REGISTER

HINLR1 is shown in Figure 4-44 and described in Table 4-44.

Figure 4-44. HINLR1 Register

31	30	29	28	27	26	25	24
AUTO_OVERRIDE	Reserved						
W-0h							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							NEST_HINT_1
							R/W-100h

7	6	5	4	3	2	1	0
NEST_HINT_1							
R/W-100h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-44. HINLR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	AUTO_OVERRIDE	W	0h	AUTO NESTING
8-0	NEST_HINT_1	R/W	100h	HOST INT 1 NESTING LEVEL

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.44 HINLR2 Register (offset = 1108h) [reset = 100h]

HOST INT 2 NESTING LEVEL REGISTER

HINLR2 is shown in Figure 4-45 and described in Table 4-45.

Figure 4-45. HINLR2 Register

31	30	29	28	27	26	25	24
AUTO_OVERRIDE	Reserved						
W-0h							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							NEST_HINT_2
							R/W-100h

7	6	5	4	3	2	1	0
NEST_HINT_2							
R/W-100h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-45. HINLR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	AUTO_OVERRIDE	W	0h	AUTO NESTING
8-0	NEST_HINT_2	R/W	100h	HOST INT 2 NESTING LEVEL

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.45 HINLR3 Register (offset = 110Ch) [reset = 100h]

HOST INT 3 NESTING LEVEL REGISTER

HINLR3 is shown in Figure 4-46 and described in Table 4-46.

Figure 4-46. HINLR3 Register

31	30	29	28	27	26	25	24
AUTO_OVERRIDE	Reserved						
W-0h							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							NEST_HINT_3
							R/W-100h

7	6	5	4	3	2	1	0
NEST_HINT_3							
R/W-100h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-46. HINLR3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	AUTO_OVERRIDE	W	0h	AUTO NESTING
8-0	NEST_HINT_3	R/W	100h	HOST INT 3 NESTING LEVEL

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.46 HINLR4 Register (offset = 1110h) [reset = 100h]

HOST INT 4 NESTING LEVEL REGISTER

HINLR4 is shown in Figure 4-47 and described in Table 4-47.

Figure 4-47. HINLR4 Register

31	30	29	28	27	26	25	24
AUTO_OVERRIDE	Reserved						
W-0h							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							NEST_HINT_4
							R/W-100h

7	6	5	4	3	2	1	0
NEST_HINT_4							
R/W-100h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-47. HINLR4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	AUTO_OVERRIDE	W	0h	AUTO NESTING
8-0	NEST_HINT_4	R/W	100h	HOST INT 4 NESTING LEVEL

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.47 HINLR5 Register (offset = 1114h) [reset = 100h]

HOST INT 5 NESTING LEVEL REGISTER

HINLR5 is shown in Figure 4-48 and described in Table 4-48.

Figure 4-48. HINLR5 Register

31	30	29	28	27	26	25	24
AUTO_OVERRIDE	Reserved						
W-0h							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							NEST_HINT_5
							R/W-100h

7	6	5	4	3	2	1	0
NEST_HINT_5							
R/W-100h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-48. HINLR5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	AUTO_OVERRIDE	W	0h	AUTO NESTING
8-0	NEST_HINT_5	R/W	100h	HOST INT 5 NESTING LEVEL

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.48 HINLR6 Register (offset = 1118h) [reset = 100h]

HOST INT 6 NESTING LEVEL REGISTER

HINLR6 is shown in Figure 4-49 and described in Table 4-49.

Figure 4-49. HINLR6 Register

31	30	29	28	27	26	25	24
AUTO_OVERRIDE	Reserved						
W-0h							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							NEST_HINT_6
							R/W-100h

7	6	5	4	3	2	1	0
NEST_HINT_6							
R/W-100h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-49. HINLR6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	AUTO_OVERRIDE	W	0h	AUTO NESTING
8-0	NEST_HINT_6	R/W	100h	HOST INT 6 NESTING LEVEL

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.49 HINLR7 Register (offset = 111Ch) [reset = 100h]

HOST INT 7 NESTING LEVEL REGISTER

HINLR7 is shown in Figure 4-50 and described in Table 4-50.

Figure 4-50. HINLR7 Register

31	30	29	28	27	26	25	24
AUTO_OVERRIDE	Reserved						
W-0h							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							NEST_HINT_7
							R/W-100h

7	6	5	4	3	2	1	0
NEST_HINT_7							
R/W-100h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-50. HINLR7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	AUTO_OVERRIDE	W	0h	AUTO NESTING
8-0	NEST_HINT_7	R/W	100h	HOST INT 7 NESTING LEVEL

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.50 HINLR8 Register (offset = 1120h) [reset = 100h]

HOST INT 8 NESTING LEVEL REGISTER

HINLR8 is shown in Figure 4-51 and described in Table 4-51.

Figure 4-51. HINLR8 Register

31	30	29	28	27	26	25	24
AUTO_OVERRIDE	Reserved						
W-0h							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							NEST_HINT_8
							R/W-100h

7	6	5	4	3	2	1	0
NEST_HINT_8							
R/W-100h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-51. HINLR8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	AUTO_OVERRIDE	W	0h	AUTO NESTING
8-0	NEST_HINT_8	R/W	100h	HOST INT 8 NESTING LEVEL

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.51 HINLR9 Register (offset = 1124h) [reset = 100h]

HOST INT 9 NESTING LEVEL REGISTER

HINLR9 is shown in Figure 4-52 and described in Table 4-52.

Figure 4-52. HINLR9 Register

31	30	29	28	27	26	25	24
AUTO_OVERRIDE	Reserved						
W-0h							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							NEST_HINT_9
							R/W-100h

7	6	5	4	3	2	1	0
NEST_HINT_9							
R/W-100h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-52. HINLR9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	AUTO_OVERRIDE	W	0h	AUTO NESTING
8-0	NEST_HINT_9	R/W	100h	HOST INT 9 NESTING LEVEL

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.52 EICR Register (offset = 2Ch) [reset = 0h]

ENABLE CLEAR INDEX REGISTER

EICR is shown in Figure 4-53 and described in Table 4-53.

Figure 4-53. EICR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																						ENABLE_CLR_INDEX									
																						W-0h									

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-53. EICR Register Field Descriptions

Bit	Field	Type	Reset	Description
9-0	ENABLE_CLR_INDEX	W	0h	ENABLE CLEAR INDEX REGISTER (WRITE INDEX TO CLEAR ENABLE OF)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.53 SRSR0 Register (offset = 200h) [reset = 0h]

RAW STATUS REGISTER 0

SRSR0 is shown in Figure 4-54 and described in Table 4-54.

Figure 4-54. SRSR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAW_STATUS_31_0																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-54. SRSR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RAW_STATUS_31_0	R/W	0h	RAW STATUS (WRITE 1 TO CLEAR) FOR INTR_IN[0]

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.54 SRSR1 Register (offset = 204h) [reset = 0h]

RAW STATUS REGISTER 1

SRSR1 is shown in Figure 4-55 and described in Table 4-55.

Figure 4-55. SRSR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAW_STATUS_63_32																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-55. SRSR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RAW_STATUS_63_32	R/W	0h	RAW STATUS (WRITE 1 TO CLEAR) FOR SLV_EVENTS_IN[0]

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.55 HIEISR Register (offset = 34h) [reset = 0h]

HOST INT ENABLE SET INDEX REGISTER

HIEISR is shown in Figure 4-56 and described in Table 4-56.

Figure 4-56. HIEISR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																						HINT_ENABLE_SET_INDEX									
																						R/W-0h									

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-56. HIEISR Register Field Descriptions

Bit	Field	Type	Reset	Description
9-0	HINT_ENABLE_SET_INDEX	R/W	0h	ENABLE SET FOR HOST INTERRUPTS

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.56 HIER Register (offset = 1500h) [reset = 0h]

HOST INT ENABLE REGISTER 0

HIER is shown in Figure 4-57 and described in Table 4-57.

Figure 4-57. HIER Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																						ENABLE_HINT									
																						R/W-0h									

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-57. HIER Register Field Descriptions

Bit	Field	Type	Reset	Description
9-0	ENABLE_HINT	R/W	0h	ENABLE FOR HOST INT

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.57 HIDISR Register (offset = 38h) [reset = 0h]

HOST INT ENABLE CLEAR INDEX REGISTER

HIDISR is shown in Figure 4-58 and described in Table 4-58.

Figure 4-58. HIDISR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																						HINT_ENABLE_CLR_INDEX									
																						R/W-0h									

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-58. HIDISR Register Field Descriptions

Bit	Field	Type	Reset	Description
9-0	HINT_ENABLE_CLR_INDEX	R/W	0h	ENABLE CLEAR FOR HOST INTERRUPTS

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.58 SECR0 Register (offset = 280h) [reset = 0h]

ENABLED STATUS REGISTER 0

SECR0 is shown in Figure 4-59 and described in Table 4-59.

Figure 4-59. SECR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENA_STATUS_31_0																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-59. SECR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ENA_STATUS_31_0	R/W	0h	ENABLED STATUS FOR INTR_IN[0]

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.59 SECR1 Register (offset = 284h) [reset = 0h]

ENABLED STATUS REGISTER 1

SECR1 is shown in Figure 4-60 and described in Table 4-60.

Figure 4-60. SECR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENA_STATUS_63_32																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-60. SECR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ENA_STATUS_63_32	R/W	0h	ENABLED STATUS FOR SLV_EVENTS_IN[0]

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.60 ESR0 Register (offset = 300h) [reset = 0h]

ENABLE SET REGISTER 0

ESR0 is shown in Figure 4-61 and described in Table 4-61.

Figure 4-61. ESR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE_SET_31_0																															
-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-61. ESR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ENABLE_SET_31_0		0h	ENABLE (SET) FOR INTR_IN[0]

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.61 ERS1 Register (offset = 304h) [reset = 0h]

ENABLE SET REGISTER 1

ERS1 is shown in Figure 4-62 and described in Table 4-62.

Figure 4-62. ERS1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE_SET_63_32																															
-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-62. ERS1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ENABLE_SET_63_32		0h	ENABLE (SET) FOR SLV_EVENTS_IN[0]

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.62 ECR0 Register (offset = 380h) [reset = 0h]

ENABLE CLEAR REGISTER 0

ECR0 is shown in Figure 4-63 and described in Table 4-63.

Figure 4-63. ECR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE_CLR_31_0																															
-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-63. ECR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ENABLE_CLR_31_0		0h	ENABLE CLEAR FOR INTR_IN[0]

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.5.63 ECR1 Register (offset = 384h) [reset = 0h]

ENABLE CLEAR REGISTER 1

ECR1 is shown in Figure 4-64 and described in Table 4-64.

Figure 4-64. ECR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE_CLR_63_32																															
-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-64. ECR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ENABLE_CLR_63_32		0h	ENABLE CLEAR FOR SLV_EVENTS_IN[0]

4.6 PRUSS_PRU_CTRL Registers

Table 4-65 lists the memory-mapped registers for the PRUSS_PRU_CTRL. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 4-65 should be considered as reserved locations and the register contents should not be modified.

Table 4-65. PRUSS_PRU_CTRL REGISTERS

Offset	Acronym	Register Description	Section
0h	CONTROL	CONTROL REGISTER	Section 13.11.30
Ch	CYCLE	CYCLE COUNT	Section 4.6.2
10h	STALL	STALL COUNT	Section 4.6.3
20h	CTBIR0	CONSTANT TABLE BLOCK INDEX REGISTER 0	Section 4.6.4
24h	CTBIR1	CONSTANT TABLE BLOCK INDEX REGISTER 1	Section 4.6.5
4h	STATUS	STATUS REGISTER	Section 6.3.5.36
28h	CTPPR0	CONSTANT TABLE PROGRAMMABLE POINTER REGISTER 0	Section 4.6.7
2Ch	CTPPR1	CONSTANT TABLE PROGRAMMABLE POINTER REGISTER 1	Section 4.6.8
8h	WAKEUP_EN	WAKEUP ENABLE REGISTER	Section 4.6.9

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.6.1 CONTROL Register (offset = 0h) [reset = 1h]

CONTROL REGISTER

CONTROL is shown in Figure 13-227 and described in Table 13-240.

Figure 4-65. CONTROL Register

31	30	29	28	27	26	25	24
PCOUNTER_RST_VAL							
R/W-0h							

23	22	21	20	19	18	17	16
PCOUNTER_RST_VAL							
R/W-0h							

15	14	13	12	11	10	9	8
RUNSTATE	BIG_ENDIAN	Reserved					SINGLE_STEP
R-0h	R-0h						R/W-0h

7	6	5	4	3	2	1	0
Reserved				COUNTER_ENABLE	SLEEPING	ENABLE	SOFT_RST_N
				R/W-0h	R/W-0h	R/W-0h	R-1h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-66. CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	PCOUNTER_RST_VAL	R/W	0h	
15	RUNSTATE	R	0h	
14	BIG_ENDIAN	R	0h	
8	SINGLE_STEP	R/W	0h	
3	COUNTER_ENABLE	R/W	0h	
2	SLEEPING	R/W	0h	
1	ENABLE	R/W	0h	
0	SOFT_RST_N	R	1h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.6.2 CYCLE Register (offset = Ch) [reset = 0h]

CYCLE COUNT

CYCLE is shown in Figure 4-66 and described in Table 4-67.

Figure 4-66. CYCLE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CYCLECOUNT																															
-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-67. CYCLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CYCLECOUNT		0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.6.3 STALL Register (offset = 10h) [reset = 0h]

STALL COUNT

STALL is shown in Figure 4-67 and described in Table 4-68.

Figure 4-67. STALL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STALLCOUNT																															
-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-68. STALL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	STALLCOUNT		0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.6.4 CTBIR0 Register (offset = 20h) [reset = 0h]

CONSTANT TABLE BLOCK INDEX REGISTER 0

CTBIR0 is shown in Figure 4-68 and described in Table 4-69.

Figure 4-68. CTBIR0 Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
C25_BLK_INDEX							
R/W-0h							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
C24_BLK_INDEX							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-69. CTBIR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
23-16	C25_BLK_INDEX	R/W	0h	
7-0	C24_BLK_INDEX	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.6.5 CTBIR1 Register (offset = 24h) [reset = 0h]

CONSTANT TABLE BLOCK INDEX REGISTER 1

CTBIR1 is shown in Figure 4-69 and described in Table 4-70.

Figure 4-69. CTBIR1 Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
C27_BLK_INDEX							
R/W-0h							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
C26_BLK_INDEX							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-70. CTBIR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
23-16	C27_BLK_INDEX	R/W	0h	
7-0	C26_BLK_INDEX	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.6.6 STATUS Register (offset = 4h) [reset = 0h]

STATUS REGISTER

STATUS is shown in Figure 6-124 and described in Table 6-138.

Figure 4-70. STATUS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																PCOUNTER															
																							R-0h								

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-71. STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	PCOUNTER	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.6.7 CTPPR0 Register (offset = 28h) [reset = 0h]

CONSTANT TABLE PROGRAMMABLE POINTER REGISTER 0

CTPPR0 is shown in Figure 4-71 and described in Table 4-72.

Figure 4-71. CTPPR0 Register

31	30	29	28	27	26	25	24
C29_POINTER							
R/W-0h							

23	22	21	20	19	18	17	16
C29_POINTER							
R/W-0h							

15	14	13	12	11	10	9	8
C28_POINTER							
R/W-0h							

7	6	5	4	3	2	1	0
C28_POINTER							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-72. CTPPR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	C29_POINTER	R/W	0h	
15-0	C28_POINTER	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.6.8 CTPPR1 Register (offset = 2Ch) [reset = 0h]

CONSTANT TABLE PROGRAMMABLE POINTER REGISTER 1

CTPPR1 is shown in Figure 4-72 and described in Table 4-73.

Figure 4-72. CTPPR1 Register

31	30	29	28	27	26	25	24
C31_POINTER							
R/W-0h							

23	22	21	20	19	18	17	16
C31_POINTER							
R/W-0h							

15	14	13	12	11	10	9	8
C30_POINTER							
R/W-0h							

7	6	5	4	3	2	1	0
C30_POINTER							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-73. CTPPR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	C31_POINTER	R/W	0h	
15-0	C30_POINTER	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.6.9 WAKEUP_EN Register (offset = 8h) [reset = 0h]

WAKEUP ENABLE REGISTER

WAKEUP_EN is shown in Figure 4-73 and described in Table 4-74.

Figure 4-73. WAKEUP_EN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BITWISE_ENABLES																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-74. WAKEUP_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	BITWISE_ENABLES	R/W	0h	

4.7 PRUSS_PRU_DEBUG Registers

Table 4-75 lists the memory-mapped registers for the PRUSS_PRU_DEBUG. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 4-75 should be considered as reserved locations and the register contents should not be modified.

Table 4-75. PRUSS_PRU_DEBUG REGISTERS

Offset	Acronym	Register Description	Section
0h	GPR0	DEBUG PRU GENERAL PURPOSE REGISTER 0	Section 4.7.1
64h	GPR25	DEBUG PRU GENERAL PURPOSE REGISTER 25	Section 4.7.2
68h	GPR26	DEBUG PRU GENERAL PURPOSE REGISTER 26	Section 4.7.3
6Ch	GPR27	DEBUG PRU GENERAL PURPOSE REGISTER 27	Section 4.7.4
70h	GPR28	DEBUG PRU GENERAL PURPOSE REGISTER 28	Section 4.7.5
74h	GPR29	DEBUG PRU GENERAL PURPOSE REGISTER 29	Section 4.7.6
Ch	GPR3	DEBUG PRU GENERAL PURPOSE REGISTER 3	Section 4.7.7
78h	GPR30	DEBUG PRU GENERAL PURPOSE REGISTER 30	Section 4.7.8
7Ch	GPR31	DEBUG PRU GENERAL PURPOSE REGISTER 31	Section 4.7.9
80h	CTR0	DEBUG PRU CONSTANTS TABLE ENTRY 0	Section 4.7.10
84h	CTR1	DEBUG PRU CONSTANTS TABLE ENTRY 1	Section 4.7.11
88h	CTR2	DEBUG PRU CONSTANTS TABLE ENTRY 2	Section 4.7.12
8Ch	CTR3	DEBUG PRU CONSTANTS TABLE ENTRY 3	Section 4.7.13
90h	CTR4	DEBUG PRU CONSTANTS TABLE ENTRY 4	Section 4.7.14
94h	CTR5	DEBUG PRU CONSTANTS TABLE ENTRY 5	Section 4.7.15
98h	CTR6	DEBUG PRU CONSTANTS TABLE ENTRY 6	Section 4.7.16
9Ch	CTR7	DEBUG PRU CONSTANTS TABLE ENTRY 7	Section 4.7.17
10h	GPR4	DEBUG PRU GENERAL PURPOSE REGISTER 4	Section 4.7.18
A0h	CTR8	DEBUG PRU CONSTANTS TABLE ENTRY 8	Section 4.7.19
A4h	CTR9	DEBUG PRU CONSTANTS TABLE ENTRY 9	Section 4.7.20
A8h	CTR10	DEBUG PRU CONSTANTS TABLE ENTRY 10	Section 4.7.21
ACh	CTR11	DEBUG PRU CONSTANTS TABLE ENTRY 11	Section 4.7.22
B0h	CTR12	DEBUG PRU CONSTANTS TABLE ENTRY 12	Section 4.7.23
B4h	CTR13	DEBUG PRU CONSTANTS TABLE ENTRY 13	Section 4.7.24
B8h	CTR14	DEBUG PRU CONSTANTS TABLE ENTRY 14	Section 4.7.25
BCh	CTR15	DEBUG PRU CONSTANTS TABLE ENTRY 15	Section 4.7.26
C0h	CTR16	DEBUG PRU CONSTANTS TABLE ENTRY 16	Section 4.7.27
C4h	CTR17	DEBUG PRU CONSTANTS TABLE ENTRY 17	Section 4.7.28

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 4-75. PRUSS_PRU_DEBUG REGISTERS (continued)

Offset	Acronym	Register Description	Section
14h	GPR5	DEBUG PRU GENERAL PURPOSE REGISTER 5	Section 4.7.29
C8h	CTR18	DEBUG PRU CONSTANTS TABLE ENTRY 18	Section 4.7.30
CCh	CTR19	DEBUG PRU CONSTANTS TABLE ENTRY 19	Section 4.7.31
D0h	CTR20	DEBUG PRU CONSTANTS TABLE ENTRY 20	Section 4.7.32
D4h	CTR21	DEBUG PRU CONSTANTS TABLE ENTRY 21	Section 4.7.33
D8h	CTR22	DEBUG PRU CONSTANTS TABLE ENTRY 22	Section 4.7.34
DCh	CTR23	DEBUG PRU CONSTANTS TABLE ENTRY 23	Section 4.7.35
E0h	CTR24	DEBUG PRU CONSTANTS TABLE ENTRY 24	Section 4.7.36
18h	GPR6	DEBUG PRU GENERAL PURPOSE REGISTER 6	Section 4.7.37
1Ch	GPR7	DEBUG PRU GENERAL PURPOSE REGISTER 7	Section 4.7.38
20h	GPR8	DEBUG PRU GENERAL PURPOSE REGISTER 8	Section 4.7.39
24h	GPR9	DEBUG PRU GENERAL PURPOSE REGISTER 9	Section 4.7.40
4h	GPR1	DEBUG PRU GENERAL PURPOSE REGISTER 1	Section 4.7.41
28h	GPR10	DEBUG PRU GENERAL PURPOSE REGISTER 10	Section 4.7.42
2Ch	GPR11	DEBUG PRU GENERAL PURPOSE REGISTER 11	Section 4.7.43
30h	GPR12	DEBUG PRU GENERAL PURPOSE REGISTER 12	Section 4.7.44
34h	GPR13	DEBUG PRU GENERAL PURPOSE REGISTER 13	Section 4.7.45
38h	GPR14	DEBUG PRU GENERAL PURPOSE REGISTER 14	Section 4.7.46
3Ch	GPR15	DEBUG PRU GENERAL PURPOSE REGISTER 15	Section 4.7.47
40h	GPR16	DEBUG PRU GENERAL PURPOSE REGISTER 16	Section 4.7.48
44h	GPR17	DEBUG PRU GENERAL PURPOSE REGISTER 17	Section 4.7.49
48h	GPR18	DEBUG PRU GENERAL PURPOSE REGISTER 18	Section 4.7.50
4Ch	GPR19	DEBUG PRU GENERAL PURPOSE REGISTER 19	Section 4.7.51
8h	GPR2	DEBUG PRU GENERAL PURPOSE REGISTER 2	Section 4.7.52
50h	GPR20	DEBUG PRU GENERAL PURPOSE REGISTER 20	Section 4.7.53
54h	GPR21	DEBUG PRU GENERAL PURPOSE REGISTER 21	Section 4.7.54
58h	GPR22	DEBUG PRU GENERAL PURPOSE REGISTER 22	Section 4.7.55
5Ch	GPR23	DEBUG PRU GENERAL PURPOSE REGISTER 23	Section 4.7.56
60h	GPR24	DEBUG PRU GENERAL PURPOSE REGISTER 24	Section 4.7.57

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.1 GPR0 Register (offset = 0h) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 0

GPR0 is shown in Figure 4-74 and described in Table 4-76.

Figure 4-74. GPR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-76. GPR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.2 GPR25 Register (offset = 64h) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 25

GPR25 is shown in Figure 4-75 and described in Table 4-77.

Figure 4-75. GPR25 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-77. GPR25 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.3 GPR26 Register (offset = 68h) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 26

GPR26 is shown in Figure 4-76 and described in Table 4-78.

Figure 4-76. GPR26 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-78. GPR26 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.4 GPR27 Register (offset = 6Ch) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 27

GPR27 is shown in Figure 4-77 and described in Table 4-79.

Figure 4-77. GPR27 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-79. GPR27 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.5 GPR28 Register (offset = 70h) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 28

GPR28 is shown in Figure 4-78 and described in Table 4-80.

Figure 4-78. GPR28 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-80. GPR28 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.6 GPR29 Register (offset = 74h) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 29

GPR29 is shown in Figure 4-79 and described in Table 4-81.

Figure 4-79. GPR29 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-81. GPR29 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.7 GPR3 Register (offset = Ch) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 3

GPR3 is shown in Figure 4-80 and described in Table 4-82.

Figure 4-80. GPR3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-82. GPR3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.8 GPR30 Register (offset = 78h) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 30

GPR30 is shown in Figure 4-81 and described in Table 4-83.

Figure 4-81. GPR30 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-83. GPR30 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.9 GPR31 Register (offset = 7Ch) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 31

GPR31 is shown in Figure 4-82 and described in Table 4-84.

Figure 4-82. GPR31 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPR31																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-84. GPR31 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GPR31	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.10 CTR0 Register (offset = 80h) [reset = 20000h]

DEBUG PRU CONSTANTS TABLE ENTRY 0

CTR0 is shown in Figure 4-83 and described in Table 4-85.

Figure 4-83. CTR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT																															
R-20000h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-85. CTR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT	R	20000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.11 CTR1 Register (offset = 84h) [reset = 48040000h]

DEBUG PRU CONSTANTS TABLE ENTRY 1

CTR1 is shown in Figure 4-84 and described in Table 4-86.

Figure 4-84. CTR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT																															
R-48040000h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-86. CTR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT	R	48040000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.12 CTR2 Register (offset = 88h) [reset = 4802A000h]

DEBUG PRU CONSTANTS TABLE ENTRY 2

CTR2 is shown in Figure 4-85 and described in Table 4-87.

Figure 4-85. CTR2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT																															
R-4802A000h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-87. CTR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT	R	4802A000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.13 CTR3 Register (offset = 8Ch) [reset = 30000h]

DEBUG PRU CONSTANTS TABLE ENTRY 3

CTR3 is shown in Figure 4-86 and described in Table 4-88.

Figure 4-86. CTR3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT																															
R-30000h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-88. CTR3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT	R	30000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.14 CTR4 Register (offset = 90h) [reset = 26000h]

DEBUG PRU CONSTANTS TABLE ENTRY 4

CTR4 is shown in Figure 4-87 and described in Table 4-89.

Figure 4-87. CTR4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT																															
R-26000h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-89. CTR4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT	R	26000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.15 CTR5 Register (offset = 94h) [reset = 48060000h]

DEBUG PRU CONSTANTS TABLE ENTRY 5

CTR5 is shown in Figure 4-88 and described in Table 4-90.

Figure 4-88. CTR5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT																															
R-48060000h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-90. CTR5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT	R	48060000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.16 CTR6 Register (offset = 98h) [reset = 48030000h]

DEBUG PRU CONSTANTS TABLE ENTRY 6

CTR6 is shown in Figure 4-89 and described in Table 4-91.

Figure 4-89. CTR6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT																															
R-48030000h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-91. CTR6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT	R	48030000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.17 CTR7 Register (offset = 9Ch) [reset = 28000h]

DEBUG PRU CONSTANTS TABLE ENTRY 7

CTR7 is shown in Figure 4-90 and described in Table 4-92.

Figure 4-90. CTR7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT																															
R-28000h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-92. CTR7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT	R	28000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.18 GPR4 Register (offset = 10h) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 4

GPR4 is shown in Figure 4-91 and described in Table 4-93.

Figure 4-91. GPR4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-93. GPR4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.19 CTR8 Register (offset = A0h) [reset = 46000000h]

DEBUG PRU CONSTANTS TABLE ENTRY 8

CTR8 is shown in Figure 4-92 and described in Table 4-94.

Figure 4-92. CTR8 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT																															
R-46000000h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-94. CTR8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT	R	46000000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.20 CTR9 Register (offset = A4h) [reset = 4A100000h]

DEBUG PRU CONSTANTS TABLE ENTRY 9

CTR9 is shown in Figure 4-93 and described in Table 4-95.

Figure 4-93. CTR9 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT																															
R-4A100000h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-95. CTR9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT	R	4A100000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.21 CTR10 Register (offset = A8h) [reset = 48318000h]

DEBUG PRU CONSTANTS TABLE ENTRY 10

CTR10 is shown in Figure 4-94 and described in Table 4-96.

Figure 4-94. CTR10 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT																															
R-48318000h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-96. CTR10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT	R	48318000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.22 CTR11 Register (offset = ACh) [reset = 48022000h]

DEBUG PRU CONSTANTS TABLE ENTRY 11

CTR11 is shown in Figure 4-95 and described in Table 4-97.

Figure 4-95. CTR11 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT																															
R-48022000h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-97. CTR11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT	R	48022000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.23 CTR12 Register (offset = B0h) [reset = 48024000h]

DEBUG PRU CONSTANTS TABLE ENTRY 12

CTR12 is shown in Figure 4-96 and described in Table 4-98.

Figure 4-96. CTR12 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT																															
R-48024000h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-98. CTR12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT	R	48024000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.24 CTR13 Register (offset = B4h) [reset = 48310000h]

DEBUG PRU CONSTANTS TABLE ENTRY 13

CTR13 is shown in Figure 4-97 and described in Table 4-99.

Figure 4-97. CTR13 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT																															
R-48310000h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-99. CTR13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT	R	48310000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.25 CTR14 Register (offset = B8h) [reset = 481CC000h]

DEBUG PRU CONSTANTS TABLE ENTRY 14

CTR14 is shown in Figure 4-98 and described in Table 4-100.

Figure 4-98. CTR14 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT																															
R-481CC000h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-100. CTR14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT	R	481CC000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.26 CTR15 Register (offset = BCh) [reset = 481D0000h]

DEBUG PRU CONSTANTS TABLE ENTRY 15

CTR15 is shown in Figure 4-99 and described in Table 4-101.

Figure 4-99. CTR15 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT																															
R-481D0000h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-101. CTR15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT	R	481D0000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.27 CTR16 Register (offset = C0h) [reset = 481A0000h]

DEBUG PRU CONSTANTS TABLE ENTRY 16

CTR16 is shown in Figure 4-100 and described in Table 4-102.

Figure 4-100. CTR16 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT																															
R-481A0000h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-102. CTR16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT	R	481A0000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.28 CTR17 Register (offset = C4h) [reset = 4819C000h]

DEBUG PRU CONSTANTS TABLE ENTRY 17

CTR17 is shown in Figure 4-101 and described in Table 4-103.

Figure 4-101. CTR17 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT																															
R-4819C000h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-103. CTR17 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT	R	4819C000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.29 GPR5 Register (offset = 14h) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 5

GPR5 is shown in Figure 4-102 and described in Table 4-104.

Figure 4-102. GPR5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-104. GPR5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.30 CTR18 Register (offset = C8h) [reset = 48300000h]

DEBUG PRU CONSTANTS TABLE ENTRY 18

CTR18 is shown in Figure 4-103 and described in Table 4-105.

Figure 4-103. CTR18 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT																															
R-48300000h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-105. CTR18 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT	R	48300000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.31 CTR19 Register (offset = CCh) [reset = 48302000h]

DEBUG PRU CONSTANTS TABLE ENTRY 19

CTR19 is shown in Figure 4-104 and described in Table 4-106.

Figure 4-104. CTR19 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT																															
R-48302000h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-106. CTR19 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT	R	48302000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.32 CTR20 Register (offset = D0h) [reset = 48304000h]

DEBUG PRU CONSTANTS TABLE ENTRY 20

CTR20 is shown in Figure 4-105 and described in Table 4-107.

Figure 4-105. CTR20 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT																															
R-48304000h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-107. CTR20 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT	R	48304000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.33 CTR21 Register (offset = D4h) [reset = 32400h]

DEBUG PRU CONSTANTS TABLE ENTRY 21

CTR21 is shown in Figure 4-106 and described in Table 4-108.

Figure 4-106. CTR21 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT																															
R-32400h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-108. CTR21 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT	R	32400h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.34 CTR22 Register (offset = D8h) [reset = 480C8000h]

DEBUG PRU CONSTANTS TABLE ENTRY 22

CTR22 is shown in Figure 4-107 and described in Table 4-109.

Figure 4-107. CTR22 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT																															
R-480C8000h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-109. CTR22 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT	R	480C8000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.35 CTR23 Register (offset = DCh) [reset = 480CA000h]

DEBUG PRU CONSTANTS TABLE ENTRY 23

CTR23 is shown in Figure 4-108 and described in Table 4-110.

Figure 4-108. CTR23 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT																															
R-480CA000h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-110. CTR23 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CT	R	480CA000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.36 CTR24 Register (offset = E0h) [reset = 0h]

DEBUG PRU CONSTANTS TABLE ENTRY 24

CTR24 is shown in Figure 4-109 and described in Table 4-111.

Figure 4-109. CTR24 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTR24																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-111. CTR24 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CTR24	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.37 GPR6 Register (offset = 18h) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 6

GPR6 is shown in Figure 4-110 and described in Table 4-112.

Figure 4-110. GPR6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-112. GPR6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.38 GPR7 Register (offset = 1Ch) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 7

GPR7 is shown in Figure 4-111 and described in Table 4-113.

Figure 4-111. GPR7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-113. GPR7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.39 GPR8 Register (offset = 20h) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 8

GPR8 is shown in Figure 4-112 and described in Table 4-114.

Figure 4-112. GPR8 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-114. GPR8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.40 GPR9 Register (offset = 24h) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 9

GPR9 is shown in Figure 4-113 and described in Table 4-115.

Figure 4-113. GPR9 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-115. GPR9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.41 GPR1 Register (offset = 4h) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 1

GPR1 is shown in Figure 4-114 and described in Table 4-116.

Figure 4-114. GPR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-116. GPR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.42 GPR10 Register (offset = 28h) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 10

GPR10 is shown in Figure 4-115 and described in Table 4-117.

Figure 4-115. GPR10 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-117. GPR10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.43 GPR11 Register (offset = 2Ch) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 11

GPR11 is shown in Figure 4-116 and described in Table 4-118.

Figure 4-116. GPR11 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-118. GPR11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.44 GPR12 Register (offset = 30h) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 12

GPR12 is shown in Figure 4-117 and described in Table 4-119.

Figure 4-117. GPR12 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-119. GPR12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.45 GPR13 Register (offset = 34h) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 13

GPR13 is shown in Figure 4-118 and described in Table 4-120.

Figure 4-118. GPR13 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-120. GPR13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.46 GPR14 Register (offset = 38h) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 14

GPR14 is shown in Figure 4-119 and described in Table 4-121.

Figure 4-119. GPR14 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-121. GPR14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.47 GPR15 Register (offset = 3Ch) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 15

GPR15 is shown in Figure 4-120 and described in Table 4-122.

Figure 4-120. GPR15 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-122. GPR15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.48 GPR16 Register (offset = 40h) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 16

GPR16 is shown in Figure 4-121 and described in Table 4-123.

Figure 4-121. GPR16 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-123. GPR16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.49 GPR17 Register (offset = 44h) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 17

GPR17 is shown in Figure 4-122 and described in Table 4-124.

Figure 4-122. GPR17 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-124. GPR17 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.50 GPR18 Register (offset = 48h) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 18

GPR18 is shown in Figure 4-123 and described in Table 4-125.

Figure 4-123. GPR18 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-125. GPR18 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.51 GPR19 Register (offset = 4Ch) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 19

GPR19 is shown in Figure 4-124 and described in Table 4-126.

Figure 4-124. GPR19 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-126. GPR19 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.52 GPR2 Register (offset = 8h) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 2

GPR2 is shown in Figure 4-125 and described in Table 4-127.

Figure 4-125. GPR2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-127. GPR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.53 GPR20 Register (offset = 50h) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 20

GPR20 is shown in Figure 4-126 and described in Table 4-128.

Figure 4-126. GPR20 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-128. GPR20 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.54 GPR21 Register (offset = 54h) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 21

GPR21 is shown in Figure 4-127 and described in Table 4-129.

Figure 4-127. GPR21 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-129. GPR21 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.55 GPR22 Register (offset = 58h) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 22

GPR22 is shown in Figure 4-128 and described in Table 4-130.

Figure 4-128. GPR22 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-130. GPR22 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.56 GPR23 Register (offset = 5Ch) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 23

GPR23 is shown in Figure 4-129 and described in Table 4-131.

Figure 4-129. GPR23 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-131. GPR23 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.7.57 GPR24 Register (offset = 60h) [reset = 0h]

DEBUG PRU GENERAL PURPOSE REGISTER 24

GPR24 is shown in Figure 4-130 and described in Table 4-132.

Figure 4-130. GPR24 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GP																															
R/W																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-132. GPR24 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	GP	R/W		

4.8 PRUSS_CFG Registers

Table 4-133 lists the memory-mapped registers for the PRUSS_CFG. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 4-133 should be considered as reserved locations and the register contents should not be modified.

Table 4-133. PRUSS_CFG REGISTERS

Offset	Acronym	Register Description	Section
0h	REVID	HL08 REVID REGISTER	Section 14.4.3.25
Ch	GPCFG1	GP CFG1	Section 4.8.2
10h	CGR	CLOCK GATING	Section 4.8.3
14h	ISRP	IRQSTATUS_RAW_PARITY	Section 4.8.4
18h	ISP	IRQSTATUS_PARITY	Section 4.8.5
1Ch	IESP	IRQENABLE_SET_PARITY	Section 4.8.6
20h	IECP	IRQENABLE_CLR_PARITY	Section 4.8.7
24h	SCRP	SCR PRIORITY CFG	Section 4.8.8
4h	SYSCFG	SYSCFG	Section 4.8.9
28h	PMAO	PRU MASTER OCP ADDRESSOFFSET	Section 4.8.10
2Ch	MII_RT	MII_RT EVENT ENABLE	Section 4.8.11
30h	IEPCLK	IEP CLOCK SOURCE	Section 4.8.12
34h	SPP	SCRATCH PAD PRIORITY AND CONFIG	Section 4.8.13
40h	PIN_MX	PIN MUX SEL	Section 4.8.14
8h	GPCFG0	GP CFG0	Section 4.8.15

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.8.1 REVID Register (offset = 0h) [reset = 47000000h]

HL08 REVID REGISTER

REVID is shown in Figure 4-131 and described in Table 4-134.

Figure 4-131. REVID Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVID																															
R-47000000h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-134. REVID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REVID	R	47000000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.8.2 GPCFG1 Register (offset = Ch) [reset = 0h]

GP CFG1

GPCFG1 is shown in Figure 4-132 and described in Table 4-135.

Figure 4-132. GPCFG1 Register

31	30	29	28	27	26	25	24
Reserved						PRU1_GPO_SH_SEL	PRU1_GPO_DIV1
						R-0h	R/W-0h

23	22	21	20	19	18	17	16
PRU1_GPO_DIV1				PRU1_GPO_DIV0			
R/W-0h				R/W-0h			

15	14	13	12	11	10	9	8
PRU1_GPO_DIV0	PRU1_GPO_MODE	PRU1_GPI_SB	PRU1_GPI_DIV1				
R/W-0h	R/W-0h	-0h	R/W-0h				

7	6	5	4	3	2	1	0
PRU1_GPI_DIV0					PRU1_GPI_CLK_MODE	PRU1_GPI_MODE	
R/W-0h					R/W-0h	R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-135. GPCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
25	PRU1_GPO_SH_SEL	R	0h	
24-20	PRU1_GPO_DIV1	R/W	0h	
19-15	PRU1_GPO_DIV0	R/W	0h	
14	PRU1_GPO_MODE	R/W	0h	
13	PRU1_GPI_SB		0h	
12-8	PRU1_GPI_DIV1	R/W	0h	
7-3	PRU1_GPI_DIV0	R/W	0h	
2	PRU1_GPI_CLK_MODE	R/W	0h	
1-0	PRU1_GPI_MODE	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.8.3 CGR Register (offset = 10h) [reset = 24924h]

CLOCK GATING

CGR is shown in Figure 4-133 and described in Table 4-136.

Figure 4-133. CGR Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved						IEP_CLK_EN	IEP_CLK_STOP_ACK
						R/W-1h	R-0h

15	14	13	12	11	10	9	8
IEP_CLK_STOP_REQ	ECAP_CLK_EN	ECAP_CLK_STOP_ACK	ECAP_CLK_STOP_REQ	UART_CLK_EN	UART_CLK_STOP_ACK	UART_CLK_STOP_REQ	INTC_CLK_EN
R/W-0h	R/W-1h	R-0h	R/W-0h	R/W-1h	R-0h	R/W-0h	R/W-1h

7	6	5	4	3	2	1	0
INTC_CLK_STOP_ACK	INTC_CLK_STOP_REQ	PRU1_CLK_EN	PRU1_CLK_STOP_ACK	PRU1_CLK_STOP_REQ	PRU0_CLK_EN	PRU0_CLK_STOP_ACK	PRU0_CLK_STOP_REQ
R-0h	R/W-0h	R/W-1h	R-0h	R/W-0h	R/W-1h	R-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-136. CGR Register Field Descriptions

Bit	Field	Type	Reset	Description
17	IEP_CLK_EN	R/W	1h	
16	IEP_CLK_STOP_ACK	R	0h	
15	IEP_CLK_STOP_REQ	R/W	0h	
14	ECAP_CLK_EN	R/W	1h	
13	ECAP_CLK_STOP_ACK	R	0h	
12	ECAP_CLK_STOP_REQ	R/W	0h	
11	UART_CLK_EN	R/W	1h	
10	UART_CLK_STOP_ACK	R	0h	
9	UART_CLK_STOP_REQ	R/W	0h	
8	INTC_CLK_EN	R/W	1h	
7	INTC_CLK_STOP_ACK	R	0h	
6	INTC_CLK_STOP_REQ	R/W	0h	
5	PRU1_CLK_EN	R/W	1h	
4	PRU1_CLK_STOP_ACK	R	0h	
3	PRU1_CLK_STOP_REQ	R/W	0h	
2	PRU0_CLK_EN	R/W	1h	
1	PRU0_CLK_STOP_ACK	R	0h	
0	PRU0_CLK_STOP_REQ	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.8.4 ISRP Register (offset = 14h) [reset = 0h]

IRQSTATUS_RAW_PARITY

ISRP is shown in Figure 4-134 and described in Table 4-137.

Figure 4-134. ISRP Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved				RAM_PE_RAW			
				R/W-0h			

15	14	13	12	11	10	9	8
PRU1_DMEM_PE_RAW				PRU1_IMEM_PE_RAW			
R/W-0h				R/W-0h			

7	6	5	4	3	2	1	0
PRU0_DMEM_PE_RAW				PRU0_IMEM_PE_RAW			
R/W-0h				R/W-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-137. ISRP Register Field Descriptions

Bit	Field	Type	Reset	Description
19-16	RAM_PE_RAW	R/W	0h	
15-12	PRU1_DMEM_PE_RAW	R/W	0h	
11-8	PRU1_IMEM_PE_RAW	R/W	0h	
7-4	PRU0_DMEM_PE_RAW	R/W	0h	
3-0	PRU0_IMEM_PE_RAW	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.8.5 ISP Register (offset = 18h) [reset = 0h]

IRQSTATUS_PARITY

ISP is shown in Figure 4-135 and described in Table 4-138.

Figure 4-135. ISP Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved				RAM_PE			
				-0h			

15	14	13	12	11	10	9	8
PRU1_DMEM_PE				PRU1_IMEM_PE			
-0h				-0h			

7	6	5	4	3	2	1	0
PRU0_DMEM_PE				PRU0_IMEM_PE			
-0h				-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-138. ISP Register Field Descriptions

Bit	Field	Type	Reset	Description
19-16	RAM_PE		0h	
15-12	PRU1_DMEM_PE		0h	
11-8	PRU1_IMEM_PE		0h	
7-4	PRU0_DMEM_PE		0h	
3-0	PRU0_IMEM_PE		0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.8.6 IESP Register (offset = 1Ch) [reset = 0h]

IRQENABLE_SET_PARITY

IESP is shown in Figure 4-136 and described in Table 4-139.

Figure 4-136. IESP Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved				RAM_PE_SET			
				R/W-0h			

15	14	13	12	11	10	9	8
PRU1_DMEM_PE_SET				PRU1_IMEM_PE_SET			
R/W-0h				R/W-0h			

7	6	5	4	3	2	1	0
PRU0_DMEM_PE_SET				PRU0_IMEM_PE_SET			
R/W-0h				R/W-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-139. IESP Register Field Descriptions

Bit	Field	Type	Reset	Description
19-16	RAM_PE_SET	R/W	0h	
15-12	PRU1_DMEM_PE_SET	R/W	0h	
11-8	PRU1_IMEM_PE_SET	R/W	0h	
7-4	PRU0_DMEM_PE_SET	R/W	0h	
3-0	PRU0_IMEM_PE_SET	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.8.7 IECP Register (offset = 20h) [reset = 0h]

IRQENABLE_CLR_PARITY

IECP is shown in Figure 4-137 and described in Table 4-140.

Figure 4-137. IECP Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
PRU1_DMEM_PE_CLR				PRU1_IMEM_PE_CLR			
R/W-0h				R/W-0h			

7	6	5	4	3	2	1	0
PRU0_DMEM_PE_CLR				PRU0_IMEM_PE_CLR			
R/W-0h				R/W-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-140. IECP Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	PRU1_DMEM_PE_CLR	R/W	0h	
11-8	PRU1_IMEM_PE_CLR	R/W	0h	
7-4	PRU0_DMEM_PE_CLR	R/W	0h	
3-0	PRU0_IMEM_PE_CLR	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.8.8 SCRP Register (offset = 24h) [reset = AA84Bh]

SCR PRIORITY CFG

SCRP is shown in Figure 4-138 and described in Table 4-141.

Figure 4-138. SCRP Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved		SCRP_18	SCRP_17	SCRP_16	SCRP_15	SCRP_14	SCRP_13
		R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-1h	R/W-0h

15	14	13	12	11	10	9	8
SCRP_12	SCRP_11	SCRP_10	SCRP_9	SCRP_8	SCRP_7	SCRP_6	SCRP_5
R/W-1h	R/W-0h	R/W-1h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
SCRP_4		SCRP_3		SCRP_2		SCRP_1	
R/W-1h		R/W-0h		R/W-2h		R/W-3h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-141. SCRP Register Field Descriptions

Bit	Field	Type	Reset	Description
21	SCRP_18	R/W	0h	
20	SCRP_17	R/W	0h	
19	SCRP_16	R/W	1h	
18	SCRP_15	R/W	0h	
17	SCRP_14	R/W	1h	
16	SCRP_13	R/W	0h	
15	SCRP_12	R/W	1h	
14	SCRP_11	R/W	0h	
13	SCRP_10	R/W	1h	
12	SCRP_9	R/W	0h	
11	SCRP_8	R/W	1h	
10	SCRP_7	R/W	0h	
9	SCRP_6	R/W	0h	
8	SCRP_5	R/W	0h	
7-6	SCRP_4	R/W	1h	
5-4	SCRP_3	R/W	0h	
3-2	SCRP_2	R/W	2h	
1-0	SCRP_1	R/W	3h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.8.9 SYSCFG Register (offset = 4h) [reset = 1Ah]

SYSCFG

SYSCFG is shown in Figure 4-139 and described in Table 4-142.

Figure 4-139. SYSCFG Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved		SUB_MWAIT	STANDBY_INIT	STANDBY_MODE		IDLE_MODE	
		R	R/W-1h	R/W-2h		R/W-2h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-142. SYSCFG Register Field Descriptions

Bit	Field	Type	Reset	Description
5	SUB_MWAIT	R		
4	STANDBY_INIT	R/W	1h	
3-2	STANDBY_MODE	R/W	2h	
1-0	IDLE_MODE	R/W	2h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.8.10 PMAO Register (offset = 28h) [reset = 0h]

PRU MASTER OCP ADDRESSOFFSET

PMAO is shown in Figure 4-140 and described in Table 4-143.

Figure 4-140. PMAO Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved						PMAO_PRU1	PMAO_PRU0
						R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-143. PMAO Register Field Descriptions

Bit	Field	Type	Reset	Description
1	PMAO_PRU1	R/W	0h	
0	PMAO_PRU0	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.8.11 MII_RT Register (offset = 2Ch) [reset = 1h]

MII_RT EVENT ENABLE

MII_RT is shown in Figure 4-141 and described in Table 4-144.

Figure 4-141. MII_RT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															MII_RT_EVENT_EN
																															R/W-1h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-144. MII_RT Register Field Descriptions

Bit	Field	Type	Reset	Description
0	MII_RT_EVENT_EN	R/W	1h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.8.12 IEPCLK Register (offset = 30h) [reset = 0h]

IEP CLOCK SOURCE

IEPCLK is shown in Figure 4-142 and described in Table 4-145.

Figure 4-142. IEPCLK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															OCP_EN
																															R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-145. IEPCLK Register Field Descriptions

Bit	Field	Type	Reset	Description
0	OCP_EN	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.8.13 SPP Register (offset = 34h) [reset = 0h]

SCRATCH PAD PRIORITY AND CONFIG

SPP is shown in Figure 4-143 and described in Table 4-146.

Figure 4-143. SPP Register

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						XFR_SHIFT_EN	PRU1_PAD_HP_EN
						R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-146. SPP Register Field Descriptions

Bit	Field	Type	Reset	Description
1	XFR_SHIFT_EN	R/W	0h	
0	PRU1_PAD_HP_EN	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.8.14 PIN_MX Register (offset = 40h) [reset = 0h]

PIN MUX SEL

PIN_MX is shown in Figure 4-144 and described in Table 4-147.

Figure 4-144. PIN_MX Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								PIN_MUX_SEL							
																								R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-147. PIN_MX Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	PIN_MUX_SEL	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.8.15 GPCFG0 Register (offset = 8h) [reset = 0h]

GP CFG0

GPCFG0 is shown in Figure 4-145 and described in Table 4-148.

Figure 4-145. GPCFG0 Register

31	30	29	28	27	26	25	24
Reserved						PRU0_GPO_SH_SEL	PRU0_GPO_DIV1
						R-0h	R/W-0h

23	22	21	20	19	18	17	16
PRU0_GPO_DIV1				PRU0_GPO_DIV0			
R/W-0h				R/W-0h			

15	14	13	12	11	10	9	8
PRU0_GPO_DIV0	PRU0_GPO_MODE	PRU0_GPI_SB	PRU0_GPI_DIV1				
R/W-0h	R/W-0h	-0h	R/W-0h				

7	6	5	4	3	2	1	0
PRU0_GPI_DIV0					PRU0_GPI_CLK_MODE	PRU0_GPI_MODE	
R/W-0h					R/W-0h	R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-148. GPCFG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
25	PRU0_GPO_SH_SEL	R	0h	
24-20	PRU0_GPO_DIV1	R/W	0h	
19-15	PRU0_GPO_DIV0	R/W	0h	
14	PRU0_GPO_MODE	R/W	0h	
13	PRU0_GPI_SB		0h	
12-8	PRU0_GPI_DIV1	R/W	0h	
7-3	PRU0_GPI_DIV0	R/W	0h	
2	PRU0_GPI_CLK_MODE	R/W	0h	
1-0	PRU0_GPI_MODE	R/W	0h	

4.9 PRUSS_UART Registers

Table 4-149 lists the memory-mapped registers for the PRUSS_UART. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 4-149 should be considered as reserved locations and the register contents should not be modified.

Table 4-149. PRUSS_UART REGISTERS

Offset	Acronym	Register Description	Section
0h	UART RBR/TBR Registers		Section 4.9.1
Ch	UART Line Control Register		Section 4.9.2
10h	UART Modem Control Register		Section 4.9.3
14h	UART Line Status Register		Section 4.9.4
18h	UART Modem Status Register		Section 4.9.5
1Ch	UART Scratch Register		Section 4.9.6
20h	UART Divisor Register (LSB)		Section 4.9.7
24h	UART Divisor Register (MSB)		Section 4.9.8
4h	UART Interrupt Enable Register		Section 4.9.9
28h	UART Peripheral ID Register		Section 4.9.10

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 4-149. PRUSS_UART REGISTERS (continued)

Offset	Acronym	Register Description	Section
30h	UART PowerManagement and Emulation Register		Section 4.9.11
34h	UART Mode Definition Register		Section 4.9.12
8h	UART Interrupt Identification Register / FIFO Control Register		Section 4.9.13

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.9.1 UART RBR/TBR Registers Register (offset = 0h) [reset = 0h]

UART RBR/TBR Registers is shown in Figure 4-146 and described in Table 4-150.

Figure 4-146. UART RBR/TBR Registers Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved						TBR.DATA	
						W-0h	

15	14	13	12	11	10	9	8
TBR.DATA							
W-0h							

7	6	5	4	3	2	1	0
RBR.DATA							
R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-150. UART RBR/TBR Registers Register Field Descriptions

Bit	Field	Type	Reset	Description
17-8	TBR.DATA	W	0h	Transmit Buffer Register
7-0	RBR.DATA	R	0h	Receive Buffer Register

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.9.2 UART Line Control Register Register (offset = Ch) [reset = 0h]

UART Line Control Register is shown in Figure 4-147 and described in Table 4-151.

Figure 4-147. UART Line Control Register Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
DLAB	BC	SP	EPS	PEN	STB	WLS1	WLS0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-151. UART Line Control Register Register Field Descriptions

Bit	Field	Type	Reset	Description
7	DLAB	R/W	0h	Divisor Latch Access Bit
6	BC	R/W	0h	Break Control
5	SP	R/W	0h	Stick Parity
4	EPS	R/W	0h	Even Parity Select
3	PEN	R/W	0h	Parity Enable
2	STB	R/W	0h	Number of Stop Bits
1	WLS1	R/W	0h	Word Length Select Bit 1
0	WLS0	R/W	0h	Word Length Select Bit 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.9.3 UART Modem Control Register Register (offset = 10h) [reset = 0h]

UART Modem Control Register is shown in Figure 4-148 and described in Table 4-152.

Figure 4-148. UART Modem Control Register Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved		AFE	LOOP	OUT2	OUT1	RTS	Reserved
		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-152. UART Modem Control Register Register Field Descriptions

Bit	Field	Type	Reset	Description
5	AFE	R/W	0h	Autoflow Control Enable
4	LOOP	R/W	0h	LOOP Bit
3	OUT2	R/W	0h	Out2 Bit
2	OUT1	R/W	0h	Out1 Bit
1	RTS	R/W	0h	Ready to Send

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.9.4 UART Line Status Register Register (offset = 14h) [reset = 60h]

UART Line Status Register is shown in Figure 4-149 and described in Table 4-153.

Figure 4-149. UART Line Status Register Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
RXFIFOE	TEMT	THRE	BI	FE	PE	OE	DR
R-0h	R-1h	R-1h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-153. UART Line Status Register Register Field Descriptions

Bit	Field	Type	Reset	Description
7	RXFIFOE	R	0h	Receiver FIFO Error
6	TEMT	R	1h	Transmitter Empty
5	THRE	R	1h	Transmitter Holding Register
4	BI	R	0h	Break Interrupt
3	FE	R	0h	Framing Error
2	PE	R	0h	Parity Error
1	OE	R	0h	Overrun Error
0	DR	R	0h	Data Ready

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.9.5 UART Modem Status Register Register (offset = 18h) [reset = 0h]

UART Modem Status Register is shown in Figure 4-150 and described in Table 4-154.

Figure 4-150. UART Modem Status Register Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
CD	RI	DSR	CTS	DCD	TERI	DDSR	DCTS
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-154. UART Modem Status Register Register Field Descriptions

Bit	Field	Type	Reset	Description
7	CD	R	0h	Carrier Detect
6	RI	R	0h	Ring Indicator
5	DSR	R	0h	Data Set Ready
4	CTS	R	0h	Clear To Send
3	DCD	R	0h	Delta Carrier Detect
2	TERI	R	0h	Trailing Edge Ring Indicator
1	DDSR	R	0h	Delta Set Ready
0	DCTS	R	0h	Delta Clear To Send

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.9.6 UART Scratch Register Register (offset = 1Ch) [reset = 0h]

UART Scratch Register is shown in Figure 4-151 and described in Table 4-155.

Figure 4-151. UART Scratch Register Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								DATA							
																								R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-155. UART Scratch Register Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	DATA	R/W	0h	Scratch Register Bits

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.9.7 UART Divisor Register (LSB) Register (offset = 20h) [reset = 0h]

UART Divisor Register (LSB) is shown in Figure 4-152 and described in Table 4-156.

Figure 4-152. UART Divisor Register (LSB) Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								DLL							
																								R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-156. UART Divisor Register (LSB) Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	DLL	R/W	0h	Divisor Latch (LSB)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.9.8 UART Divisor Register (MSB) Register (offset = 24h) [reset = 0h]

UART Divisor Register (MSB) is shown in Figure 4-153 and described in Table 4-157.

Figure 4-153. UART Divisor Register (MSB) Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								DLH							
																								R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-157. UART Divisor Register (MSB) Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	DLH	R/W	0h	Divisor Latch (MSB)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.9.9 UART Interrupt Enable Register Register (offset = 4h) [reset = 0h]

UART Interrupt Enable Register is shown in Figure 4-154 and described in Table 4-158.

Figure 4-154. UART Interrupt Enable Register Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved				EDSSI	ELSI	ETBEI	ERBI
				R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-158. UART Interrupt Enable Register Register Field Descriptions

Bit	Field	Type	Reset	Description
3	EDSSI	R/W	0h	Enable for Modem Status Interrupt
2	ELSI	R/W	0h	Enable for Receiver Line Status Interrupt
1	ETBEI	R/W	0h	Enable for Transmitter Holding Register Empty Interrupt
0	ERBI	R/W	0h	Enable for Receiver Data Available Interrupt

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.9.10 UART Peripheral ID Register Register (offset = 28h) [reset = 44141102h]

UART Peripheral ID Register is shown in Figure 4-155 and described in Table 4-159.

Figure 4-155. UART Peripheral ID Register Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PID																															
R-44141102h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-159. UART Peripheral ID Register Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PID	R	44141102h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.9.11 UART PowerManagement and Emulation Register Register (offset = 30h) [reset = 0h]

UART PowerManagement and Emulation Register is shown in Figure 4-156 and described in Table 4-160.

Figure 4-156. UART PowerManagement and Emulation Register Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved	UTRST	URRST	Reserved				
	R/W-0h	R/W-0h					

7	6	5	4	3	2	1	0
Reserved							FREE
							R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-160. UART PowerManagement and Emulation Register Register Field Descriptions

Bit	Field	Type	Reset	Description
14	UTRST	R/W	0h	UART Transmitter Reset Bit
13	URRST	R/W	0h	UART Receiver Reset Bit
0	FREE	R/W	0h	Free Bit

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.9.12 UART Mode Definition Register Register (offset = 34h) [reset = 0h]

UART Mode Definition Register is shown in Figure 4-157 and described in Table 4-161.

Figure 4-157. UART Mode Definition Register Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															OSM_SEL
																															R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-161. UART Mode Definition Register Register Field Descriptions

Bit	Field	Type	Reset	Description
0	OSM_SEL	R/W	0h	Oversampling Mode Select

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.9.13 UART Interrupt Identification Register / FIFO Control Register Register (offset = 8h) [reset = 1h]

UART Interrupt Identification Register / FIFO Control Register is shown in Figure 4-158 and described in Table 4-162.

Figure 4-158. UART Interrupt Identification Register / FIFO Control Register Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
FCR.RXFIFTL		Reserved		FCR.DMAMODE1	FCR.TXCLR	FCR.RXCLR	FCR.FIFOEN
W-0h				W-0h	W-0h	W-0h	W-0h

7	6	5	4	3	2	1	0
IIR.FIFOEN		Reserved		IIR.INTID			IIR.IPEND
R-0h				R-0h			R-1h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-162. UART Interrupt Identification Register / FIFO Control Register Register Field Descriptions

Bit	Field	Type	Reset	Description
15-14	FCR.RXFIFTL	W	0h	Receiver Trigger Level
11	FCR.DMAMODE1	W	0h	DMA Mode Select
10	FCR.TXCLR	W	0h	Transmitter FIFO Reset
9	FCR.RXCLR	W	0h	Receiver FIFO Reset
8	FCR.FIFOEN	W	0h	FIFO Enable Register
7-6	IIR.FIFOEN	R	0h	FIFOs enabled
3-1	IIR.INTID	R	0h	Interrupt Type
0	IIR.IPEND	R	1h	Receiver Data Available Interrupt Pending

4.10 PRUSS_IEP Registers

Table 4-163 lists the memory-mapped registers for the PRUSS_IEP. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 4-163 should be considered as reserved locations and the register contents should not be modified.

Table 4-163. PRUSS_IEP REGISTERS

Offset	Acronym	Register Description	Section
0h	GLOBAL_CFG	GLOBAL CFG	Section 4.10.1
64h	CMP7	COMPARE7	Section 4.10.2
Ch	COUNT	COUNTER	Section 4.10.3
80h	RXIPG0	RXIPG0	Section 4.10.4
84h	RXIPG1	RXIPG1	Section 4.10.5
10h	CAP_CFG	CAPTURE CFG	Section 4.10.6
14h	CAP_STATUS	CAPTURE STATUS CFG	Section 4.10.7
18h	CAPR0	CAPTURE RISE0	Section 4.10.8
100h	SYNC_CTRL	SYNC CTRL	Section 4.10.9
104h	SYNC_FIRST_STAT	SYNC CTRL	Section 4.10.10
108h	SYNC0_STAT	SYNC CTRL	Section 4.10.11
10Ch	SYNC1_STAT	SYNC CTRL	Section 4.10.12

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 4-163. PRUSS_IEP REGISTERS (continued)

Offset	Acronym	Register Description	Section
110h	SYNC_PWIDTH	SYNC CTRL	Section 4.10.13
114h	SYNC0_PERIOD	SYNC CTRL	Section 4.10.14
1Ch	CAPR1	CAPTURE RISE1	Section 4.10.15
118h	SYNC1_DELAY	SYNC CTRL	Section 4.10.16
11Ch	SYNC_START	SYNC CTRL	Section 4.10.17
20h	CAPR2	CAPTURE RISE2	Section 4.10.18
24h	CAPR3	CAPTURE RISE3	Section 4.10.19
4h	GLOBAL_STATUS	STATUS	Section 4.10.20
28h	CAPR4	CAPTURE RISE4	Section 4.10.21
2Ch	CAPR5	CAPTURE RISE5	Section 4.10.22
30h	CAPR6	CAPTURE RISE6	Section 4.10.23
200h	WD_PREDIV	WD	Section 4.10.24
204h	PDI_WD_TIM	WD	Section 4.10.25
34h	CAPF6	CAPTURE FALL6	Section 4.10.26
208h	PD_WD_TIM	WD	Section 4.10.27
20Ch	WD_STATUS	WD	Section 4.10.28
210h	WD_EXP_CNT	WD	Section 4.10.29
214h	WD_CTRL	WD	Section 4.10.30
38h	CAPR7	CAPTURE RISE7	Section 4.10.31
3Ch	CAPF7	CAPTURE FALL7	Section 4.10.32
40h	CMP_CFG	COMPARE CFG	Section 4.10.33
44h	CMP_STATUS	COMPARE STATUS	Section 4.10.34
48h	CMP0	COMPARE0	Section 4.10.35
4Ch	CMP1	COMPARE1	Section 4.10.36
300h	DIGIO_CTRL	DIGIO	Section 4.10.37
304h	DIGIO_STATUS	DIGIO	Section 4.10.38
308h	DIGIO_DATA_IN	DIGIO	Section 4.10.39
30Ch	DIGIO_DATA_IN_RAW	DIGIO	Section 4.10.40
310h	DIGIO_DATA_OUT	DIGIO	Section 4.10.41
314h	DIGIO_DATA_OUT_EN	DIGIO	Section 4.10.42
318h	DIGIO_EXP	DIGIO	Section 4.10.43
8h	COMPEN	COMPENSATION	Section 4.10.44
50h	CMP2	COMPARE2	Section 4.10.45
54h	CMP3	COMPARE3	Section 4.10.46
58h	CMP4	COMPARE4	Section 4.10.47
5Ch	CMP5	COMPARE5	Section 4.10.48
60h	CMP6	COMPARE6	Section 4.10.49

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.1 GLOBAL_CFG Register (offset = 0h) [reset = 550h]

GLOBAL CFG

GLOBAL_CFG is shown in Figure 4-159 and described in Table 4-164.

Figure 4-159. GLOBAL_CFG Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved				CMP_INC			
				R/W-5h			

15	14	13	12	11	10	9	8
CMP_INC							
R/W-5h							

7	6	5	4	3	2	1	0
DEFAULT_INC				Reserved			CNT_ENABLE
R/W-5h							R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-164. GLOBAL_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
19-8	CMP_INC	R/W	5h	
7-4	DEFAULT_INC	R/W	5h	
0	CNT_ENABLE	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.2 CMP7 Register (offset = 64h) [reset = 0h]

COMPARE7

CMP7 is shown in Figure 4-160 and described in Table 4-165.

Figure 4-160. CMP7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP7																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-165. CMP7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CMP7	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.3 COUNT Register (offset = Ch) [reset = 0h]

COUNTER

COUNT is shown in Figure 4-161 and described in Table 4-166.

Figure 4-161. COUNT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNT																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-166. COUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.4 RXIPG0 Register (offset = 80h) [reset = FFFF0000h]

RXIPG0

RXIPG0 is shown in Figure 4-162 and described in Table 4-167.

Figure 4-162. RXIPG0 Register

31	30	29	28	27	26	25	24
RX_MIN_IPG							
R/W-FFFFh							

23	22	21	20	19	18	17	16
RX_MIN_IPG							
R/W-FFFFh							

15	14	13	12	11	10	9	8
RX_IPG							
R-0h							

7	6	5	4	3	2	1	0
RX_IPG							
R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-167. RXIPG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RX_MIN_IPG	R/W	FFFFh	
15-0	RX_IPG	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.5 RXIPG1 Register (offset = 84h) [reset = FFFF0000h]

RXIPG1

RXIPG1 is shown in Figure 4-163 and described in Table 4-168.

Figure 4-163. RXIPG1 Register

31	30	29	28	27	26	25	24
RX_MIN_IPG							
R/W-FFFFh							

23	22	21	20	19	18	17	16
RX_MIN_IPG							
R/W-FFFFh							

15	14	13	12	11	10	9	8
RX_IPG							
R-0h							

7	6	5	4	3	2	1	0
RX_IPG							
R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-168. RXIPG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RX_MIN_IPG	R/W	FFFFh	
15-0	RX_IPG	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.6 CAP_CFG Register (offset = 10h) [reset = 1FC00h]

CAPTURE CFG

CAP_CFG is shown in Figure 4-164 and described in Table 4-169.

Figure 4-164. CAP_CFG Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved						CAP_ASYNC_EN	
						R/W-7Fh	

15	14	13	12	11	10	9	8
CAP_ASYNC_EN						CAP7F_1ST_EVENT_EN	CAP7R_1ST_EVENT_EN
R/W-7Fh						R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
CAP6F_1ST_EVENT_EN	CAP6R_1ST_EVENT_EN	CAP_1ST_EVENT_EN					
R/W-0h	R/W-0h	R/W-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-169. CAP_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
17-10	CAP_ASYNC_EN	R/W	7Fh	
9	CAP7F_1ST_EVENT_EN	R/W	0h	
8	CAP7R_1ST_EVENT_EN	R/W	0h	
7	CAP6F_1ST_EVENT_EN	R/W	0h	
6	CAP6R_1ST_EVENT_EN	R/W	0h	
5-0	CAP_1ST_EVENT_EN	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.7 CAP_STATUS Register (offset = 14h) [reset = 0h]

CAPTURE STATUS CFG

CAP_STATUS is shown in Figure 4-165 and described in Table 4-170.

Figure 4-165. CAP_STATUS Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
CAP_RAW							
R							

15	14	13	12	11	10	9	8
Reserved					CAP_VALID	CAPF7_VALID	CAPR7_VALID
					R-0h	R-0h	R-0h

7	6	5	4	3	2	1	0
CAPF6_VALID	CAPR6_VALID	CAPR_VALID					
R-0h	R-0h				R-0h		

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-170. CAP_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
23-16	CAP_RAW	R		
10	CAP_VALID	R	0h	
9	CAPF7_VALID	R	0h	
8	CAPR7_VALID	R	0h	
7	CAPF6_VALID	R	0h	
6	CAPR6_VALID	R	0h	
5-0	CAPR_VALID	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.8 CAPR0 Register (offset = 18h) [reset = 0h]

CAPTURE RISE0

CAPR0 is shown in Figure 4-166 and described in Table 4-171.

Figure 4-166. CAPR0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPR0																															
R-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-171. CAPR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAPR0	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.9 SYNC_CTRL Register (offset = 100h) [reset = 0h]

SYNC CTRL

SYNC_CTRL is shown in Figure 4-167 and described in Table 4-172.

Figure 4-167. SYNC_CTRL Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							SYNC1_IND_EN
							R/W-0h

7	6	5	4	3	2	1	0
SYNC1_CYCLIC_EN	SYNC1_ACK_EN	SYNC0_CYCLIC_EN	SYNC0_ACK_EN	Reserved	SYNC1_EN	SYNC0_EN	SYNC_EN
R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-172. SYNC_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
8	SYNC1_IND_EN	R/W	0h	
7	SYNC1_CYCLIC_EN	R/W	0h	
6	SYNC1_ACK_EN	R/W	0h	
5	SYNC0_CYCLIC_EN	R/W	0h	
4	SYNC0_ACK_EN	R/W	0h	
2	SYNC1_EN	R/W	0h	
1	SYNC0_EN	R/W	0h	
0	SYNC_EN	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.10 SYNC_FIRST_STAT Register (offset = 104h) [reset = 0h]

SYNC CTRL

SYNC_FIRST_STAT is shown in Figure 4-168 and described in Table 4-173.

Figure 4-168. SYNC_FIRST_STAT Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved						FIRST_SYNC1	FIRST_SYNC0
						R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-173. SYNC_FIRST_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
1	FIRST_SYNC1	R	0h	
0	FIRST_SYNC0	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.11 SYNC0_STAT Register (offset = 108h) [reset = 0h]

SYNC CTRL

SYNC0_STAT is shown in Figure 4-169 and described in Table 4-174.

Figure 4-169. SYNC0_STAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															SYNC0_PEND
																															-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-174. SYNC0_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
0	SYNC0_PEND		0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.12 SYNC1_STAT Register (offset = 10Ch) [reset = 0h]

SYNC CTRL

SYNC1_STAT is shown in Figure 4-170 and described in Table 4-175.

Figure 4-170. SYNC1_STAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															SYNC1_PEND
																															-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-175. SYNC1_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
0	SYNC1_PEND		0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.13 SYNC_PWIDTH Register (offset = 110h) [reset = 0h]

SYNC CTRL

SYNC_PWIDTH is shown in Figure 4-171 and described in Table 4-176.

Figure 4-171. SYNC_PWIDTH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNC_HPW																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-176. SYNC_PWIDTH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SYNC_HPW	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.14 SYNC0_PERIOD Register (offset = 114h) [reset = 1h]

SYNC CTRL

SYNC0_PERIOD is shown in Figure 4-172 and described in Table 4-177.

Figure 4-172. SYNC0_PERIOD Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNC0_PERIOD																															
R/W-1h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-177. SYNC0_PERIOD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SYNC0_PERIOD	R/W	1h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.15 CAPR1 Register (offset = 1Ch) [reset = 0h]

CAPTURE RISE1

CAPR1 is shown in Figure 4-173 and described in Table 4-178.

Figure 4-173. CAPR1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPR1																															
R-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-178. CAPR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAPR1	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.16 SYNC1_DELAY Register (offset = 118h) [reset = 0h]

SYNC CTRL

SYNC1_DELAY is shown in Figure 4-174 and described in Table 4-179.

Figure 4-174. SYNC1_DELAY Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNC1_DELAY																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-179. SYNC1_DELAY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SYNC1_DELAY	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.17 SYNC_START Register (offset = 11Ch) [reset = 0h]

SYNC CTRL

SYNC_START is shown in Figure 4-175 and described in Table 4-180.

Figure 4-175. SYNC_START Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNC_START																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-180. SYNC_START Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SYNC_START	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.18 CAPR2 Register (offset = 20h) [reset = 0h]

CAPTURE RISE2

CAPR2 is shown in Figure 4-176 and described in Table 4-181.

Figure 4-176. CAPR2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPR2																															
R-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-181. CAPR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAPR2	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.19 CAPR3 Register (offset = 24h) [reset = 0h]

CAPTURE RISE3

CAPR3 is shown in Figure 4-177 and described in Table 4-182.

Figure 4-177. CAPR3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPR3																															
R-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-182. CAPR3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAPR3	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.20 GLOBAL_STATUS Register (offset = 4h) [reset = 0h]

STATUS

GLOBAL_STATUS is shown in Figure 4-178 and described in Table 4-183.

Figure 4-178. GLOBAL_STATUS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															CNT_OVF
																															-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-183. GLOBAL_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
0	CNT_OVF		0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.21 CAPR4 Register (offset = 28h) [reset = 0h]

CAPTURE RISE4

CAPR4 is shown in Figure 4-179 and described in Table 4-184.

Figure 4-179. CAPR4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPR4																															
R-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-184. CAPR4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAPR4	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.22 CAPR5 Register (offset = 2Ch) [reset = 0h]

CAPTURE RISE5

CAPR5 is shown in Figure 4-180 and described in Table 4-185.

Figure 4-180. CAPR5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPR5																															
R-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-185. CAPR5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAPR5	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.23 CAPR6 Register (offset = 30h) [reset = 0h]

CAPTURE RISE6

CAPR6 is shown in Figure 4-181 and described in Table 4-186.

Figure 4-181. CAPR6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPR6																															
R-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-186. CAPR6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAPR6	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.24 WD_PREDIV Register (offset = 200h) [reset = 4E20h]

WD

WD_PREDIV is shown in Figure 4-182 and described in Table 4-187.

Figure 4-182. WD_PREDIV Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																PRE_DIV															
																							R/W-4E20h								

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-187. WD_PREDIV Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	PRE_DIV	R/W	4E20h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.25 PDI_WD_TIM Register (offset = 204h) [reset = 3E8h]

WD

PDI_WD_TIM is shown in Figure 4-183 and described in Table 4-188.

Figure 4-183. PDI_WD_TIM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																PDI_WD_TIME															
																						R/W-3E8h									

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-188. PDI_WD_TIM Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	PDI_WD_TIME	R/W	3E8h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.26 CAPF6 Register (offset = 34h) [reset = 0h]

CAPTURE FALL6

CAPF6 is shown in Figure 4-184 and described in Table 4-189.

Figure 4-184. CAPF6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPF6																															
R-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-189. CAPF6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAPF6	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.27 PD_WD_TIM Register (offset = 208h) [reset = 3E8h]

WD

PD_WD_TIM is shown in Figure 4-185 and described in Table 4-190.

Figure 4-185. PD_WD_TIM Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																PD_WD_TIME															
																						R/W-3E8h									

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-190. PD_WD_TIM Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	PD_WD_TIME	R/W	3E8h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.28 WD_STATUS Register (offset = 20Ch) [reset = 10001h]

WD

WD_STATUS is shown in Figure 4-186 and described in Table 4-191.

Figure 4-186. WD_STATUS Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							PDI_WD_STAT
							R-1h

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved							PD_WD_STAT
							R-1h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-191. WD_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
16	PDI_WD_STAT	R	1h	
0	PD_WD_STAT	R	1h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.29 WD_EXP_CNT Register (offset = 210h) [reset = 0h]

WD

WD_EXP_CNT is shown in Figure 4-187 and described in Table 4-192.

Figure 4-187. WD_EXP_CNT Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
PD_EXP_CNT							
R/W-0h							

7	6	5	4	3	2	1	0
PDI_EXP_CNT							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-192. WD_EXP_CNT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	PD_EXP_CNT	R/W	0h	
7-0	PDI_EXP_CNT	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.30 WD_CTRL Register (offset = 214h) [reset = 0h]

WD

WD_CTRL is shown in Figure 4-188 and described in Table 4-193.

Figure 4-188. WD_CTRL Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							PDI_WD_EN
							R/W-0h

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved							PD_WD_EN
							R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-193. WD_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
16	PDI_WD_EN	R/W	0h	
0	PD_WD_EN	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.31 CAPR7 Register (offset = 38h) [reset = 0h]

CAPTURE RISE7

CAPR7 is shown in Figure 4-189 and described in Table 4-194.

Figure 4-189. CAPR7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPR7																															
R-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-194. CAPR7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAPR7	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.32 CAPF7 Register (offset = 3Ch) [reset = 0h]

CAPTURE FALL7

CAPF7 is shown in Figure 4-190 and described in Table 4-195.

Figure 4-190. CAPF7 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPF7																															
R-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-195. CAPF7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAPF7	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.33 CMP_CFG Register (offset = 40h) [reset = 0h]

COMPARE CFG

CMP_CFG is shown in Figure 4-191 and described in Table 4-196.

Figure 4-191. CMP_CFG Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							CMP_EN
							R/W-0h

7	6	5	4	3	2	1	0
CMP_EN							CMP0_RST_CNT_EN
R/W-0h							R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-196. CMP_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
8-1	CMP_EN	R/W	0h	
0	CMP0_RST_CNT_EN	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.34 CMP_STATUS Register (offset = 44h) [reset = 0h]

COMPARE STATUS

CMP_STATUS is shown in Figure 4-192 and described in Table 4-197.

Figure 4-192. CMP_STATUS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								CMP_HIT							
																								-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-197. CMP_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	CMP_HIT		0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.35 CMP0 Register (offset = 48h) [reset = 0h]

COMPARE0

CMP0 is shown in Figure 4-193 and described in Table 4-198.

Figure 4-193. CMP0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP0																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-198. CMP0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CMP0	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.36 CMP1 Register (offset = 4Ch) [reset = 0h]

COMPARE1

CMP1 is shown in Figure 4-194 and described in Table 4-199.

Figure 4-194. CMP1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP1																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-199. CMP1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CMP1	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.37 DIGIO_CTRL Register (offset = 300h) [reset = 4h]

DIGIO

DIGIO_CTRL is shown in Figure 4-195 and described in Table 4-200.

Figure 4-195. DIGIO_CTRL Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
OUT_MODE		IN_MODE		WD_MODE	BIDI_MODE	OUTVALID_MODE	OUTVALID_POL
R/W-0h		R/W-0h		R/W-0h	R-1h	R/W-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-200. DIGIO_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
7-6	OUT_MODE	R/W	0h	
5-4	IN_MODE	R/W	0h	
3	WD_MODE	R/W	0h	
2	BIDI_MODE	R	1h	
1	OUTVALID_MODE	R/W	0h	
0	OUTVALID_POL	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.38 DIGIO_STATUS Register (offset = 304h) [reset = 0h]

DIGIO

DIGIO_STATUS is shown in Figure 4-196 and described in Table 4-201.

Figure 4-196. DIGIO_STATUS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIGIO_STAT																															
R-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-201. DIGIO_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DIGIO_STAT	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.39 DIGIO_DATA_IN Register (offset = 308h) [reset = 0h]

DIGIO

DIGIO_DATA_IN is shown in Figure 4-197 and described in Table 4-202.

Figure 4-197. DIGIO_DATA_IN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_IN																															
R																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-202. DIGIO_DATA_IN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA_IN	R		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.40 DIGIO_DATA_IN_RAW Register (offset = 30Ch) [reset = 0h]

DIGIO

DIGIO_DATA_IN_RAW is shown in Figure 4-198 and described in Table 4-203.

Figure 4-198. DIGIO_DATA_IN_RAW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_IN_RAW																															
R																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-203. DIGIO_DATA_IN_RAW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA_IN_RAW	R		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.41 DIGIO_DATA_OUT Register (offset = 310h) [reset = 0h]

DIGIO

DIGIO_DATA_OUT is shown in Figure 4-199 and described in Table 4-204.

Figure 4-199. DIGIO_DATA_OUT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_OUT																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-204. DIGIO_DATA_OUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA_OUT	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.42 DIGIO_DATA_OUT_EN Register (offset = 314h) [reset = 0h]

DIGIO

DIGIO_DATA_OUT_EN is shown in Figure 4-200 and described in Table 4-205.

Figure 4-200. DIGIO_DATA_OUT_EN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_OUT_EN																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-205. DIGIO_DATA_OUT_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DATA_OUT_EN	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.43 DIGIO_EXP Register (offset = 318h) [reset = 20h]

DIGIO

DIGIO_EXP is shown in Figure 4-201 and described in Table 4-206.

Figure 4-201. DIGIO_EXP Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved		EOF_SEL	SOF_SEL	SOF_DLY			
		R/W-0h	R/W-0h	R/W-0h			

7	6	5	4	3	2	1	0
OUTVALID_DLY				Reserved	SW_OUTVALID	OUTVALID_OVR_EN	SW_DATA_OUT_UPDATE
R/W-2h					R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-206. DIGIO_EXP Register Field Descriptions

Bit	Field	Type	Reset	Description
13	EOF_SEL	R/W	0h	
12	SOF_SEL	R/W	0h	
11-8	SOF_DLY	R/W	0h	
7-4	OUTVALID_DLY	R/W	2h	
2	SW_OUTVALID	R/W	0h	
1	OUTVALID_OVR_EN	R/W	0h	
0	SW_DATA_OUT_UPDATE	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.44 COMPEN Register (offset = 8h) [reset = 0h]

COMPENSATION

COMPEN is shown in Figure 4-202 and described in Table 4-207.

Figure 4-202. COMPEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									COMPEN_CNT																						
																			R/W-0h												

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-207. COMPEN Register Field Descriptions

Bit	Field	Type	Reset	Description
22-0	COMPEN_CNT	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.45 CMP2 Register (offset = 50h) [reset = 0h]

COMPARE2

CMP2 is shown in Figure 4-203 and described in Table 4-208.

Figure 4-203. CMP2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP2																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-208. CMP2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CMP2	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.46 CMP3 Register (offset = 54h) [reset = 0h]

COMPARE3

CMP3 is shown in Figure 4-204 and described in Table 4-209.

Figure 4-204. CMP3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP3																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-209. CMP3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CMP3	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.47 CMP4 Register (offset = 58h) [reset = 0h]

COMPARE4

CMP4 is shown in Figure 4-205 and described in Table 4-210.

Figure 4-205. CMP4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP4																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-210. CMP4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CMP4	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.48 CMP5 Register (offset = 5Ch) [reset = 0h]

COMPARE5

CMP5 is shown in Figure 4-206 and described in Table 4-211.

Figure 4-206. CMP5 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP5																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-211. CMP5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CMP5	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.10.49 CMP6 Register (offset = 60h) [reset = 0h]

COMPARE6

CMP6 is shown in Figure 4-207 and described in Table 4-212.

Figure 4-207. CMP6 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP6																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-212. CMP6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CMP6	R/W	0h	

4.11 PRUSS_ECAP Registers

Table 4-213 lists the memory-mapped registers for the PRUSS_ECAP. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 4-213 should be considered as reserved locations and the register contents should not be modified.

Table 4-213. PRUSS_ECAP REGISTERS

Offset	Acronym	Register Description	Section
0h	TSCNT	TIME STAMP COUNTER REGISTER	Section 4.11.1
Ch	CAP2	CAPTURE-2 REGISTER	Section 14.3.4.4
10h	CAP3	CAPTURE-3 REGISTER	Section 14.3.4.5
14h	CAP4	CAPTURE-4 REGISTER	Section 14.3.4.6
26h	ECCTL1	ECAP CONTROL REGISTER 1	Section 14.3.4.7
4h	CNTPHS	COUNTER PHASE CONTROL REGISTER	Section 4.11.6
28h	ECCTL2	ECAP CONTROL REGISTER 2	Section 14.3.4.8
2Ah	ECEINT	ECAP INTERRUPT ENABLE REGISTER	Section 14.3.4.9
2Ch	ECFLG	ECAP INTERRUPT FLAG REGISTER	Section 14.3.4.10
30h	ECCLR	ECAP INTERRUPT CLEAR REGISTER	Section 14.3.4.11
34h	ECFRC	ECAP INTERRUPT FORCING REGISTER	Section 14.3.4.12
8h	CAP1	CAPTURE-1 REGISTER	Section 14.3.4.3
5Ch	PID	ECAP PERIPHERAL ID REGISTER	Section 21.4.1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.11.1 TSCNT Register (offset = 0h) [reset = 0h]

TIME STAMP COUNTER REGISTER

TSCNT is shown in Figure 4-208 and described in Table 4-214.

Figure 4-208. TSCNT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSCNT																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-214. TSCNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TSCNT	R/W	0h	ACTIVE 32 BIT COUNTER REGISTER WHICH IS USED AS THE CAPTURE TIME-BASE

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.11.2 CAP2 Register (offset = Ch) [reset = 0h]

CAPTURE-2 REGISTER

CAP2 is shown in Figure 4-209 and described in Table 4-215.

Figure 4-209. CAP2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAP2																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-215. CAP2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAP2	R/W	0h	THIS REGISTER CAN BE LOADED (WRITTEN) BY :TIME-STAMP (I.E. COUNTER VALUE) DURING A CAPTURE EVENTS/W MAY BE USEFUL FOR TEST PURPOSESACMP SHADOW REGISTER (I.E. CAP4) WHEN USED IN APWM MODE

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.11.3 CAP3 Register (offset = 10h) [reset = 0h]

CAPTURE-3 REGISTER

CAP3 is shown in Figure 4-210 and described in Table 4-216.

Figure 4-210. CAP3 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAP3																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-216. CAP3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAP3	R/W	0h	IN CMP MODE THIS IS A TIME-STAMP CAPTURE REGISTERIN APMW MODE THIS IS THE PERIOD SHADOW (APER) REGISTER. USER UPDATES THE PWM PERIOD VALUE VIA THIS REGISTER. IN THIS MODE CAP3 (APRD) SHADOWS CAP1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.11.4 CAP4 Register (offset = 14h) [reset = 0h]

CAPTURE-4 REGISTER

CAP4 is shown in Figure 4-211 and described in Table 4-217.

Figure 4-211. CAP4 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAP4																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-217. CAP4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAP4	R/W	0h	IN CMP MODE THIS IS A TIME-STAMP CAPTURE REGISTERIN APMW MODE THIS IS THE COMPARE SHADOW (ACMP) REGISTER. USER UPDATES THE PWM COMPARE VALUE VIA THIS REGISTER. IN THIS MODE CAP4 (ACMP) SHADOWS CAP2

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.11.5 ECCTL1 Register (offset = 26h) [reset = 0h]

ECAP CONTROL REGISTER 1

ECCTL1 is shown in Figure 4-212 and described in Table 4-218.

Figure 4-212. ECCTL1 Register

15	14	13	12	11	10	9	8
FREE	SOFT	EVTFLTPS					CAPLDEN
R/W-0h	R/W-0h	R/W-0h					R/W-0h
7	6	5	4	3	2	1	0
CTRRST4	CAP4POL	CTRRST3	CAP3POL	CTRRST2	CAP2POL	CTRRST1	CAP1POL
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-218. ECCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
15	FREE	R/W	0h	EMULATION CONTROL0,0 TSCNT COUNTER STOPS IMMEDIATELY ON EMULATION SUSPEND0,1 TSCNT COUNTER RUNS UNTIL = 01,X TSCNT COUNTER IS UNAFFECTED BY EMULATION SUSPEND (RUN FREE)
14	SOFT	R/W	0h	EMULATION CONTROL0,0 TSCNT COUNTER STOPS IMMEDIATELY ON EMULATION SUSPEND0,1 TSCNT COUNTER RUNS UNTIL = 01,X TSCNT COUNTER IS UNAFFECTED BY EMULATION SUSPEND (RUN FREE)
13-9	EVTFLTPS	R/W	0h	EVENT FILTER PRESCALE SELECT:0,0,0,0,0DIVIDE BY 1 (I.E. NO PRESCALE, BY-PASS THE PRESCALER)0,0,0,0,1DIVIDE BY 20,0,0,1,0DIVIDE BY 40,0,0,1,1DIVIDE BY 60,0,1,0,0DIVIDE BY 80,0,1,0,1DIVIDE BY 10. . . . .1,1,1,1,0DIVIDE BY 601,1,1,1,1DIVIDE BY 62
8	CAPLDEN	R/W	0h	ENABLE LOADING OF CAP1-4 REGISTERS ON A CAPTURE EVENT:0DISABLE CAP1-4 REGISTER LOADS AT CAPTURE EVENT TIME1ENABLE CAP1-4 REGISTER LOADS AT CAPTURE EVENT TIME
7	CTRRST4	R/W	0h	COUNTER RESET ON CAPTURE EVENT 4:0DO NOT RESET COUNTER ON CAPTURE EVENT 4 (ABSOLUTE TIME STAMP)1RESET COUNTER AFTER EVENT 4 TIME-STAMP HAS BEEN CAPTURED(USED IN DIFFERENCE MODE OPERATION)
6	CAP4POL	R/W	0h	CAPTURE EVENT 4 POLARITY SELECT:0CAPTURE EVENT 4 TRIGGERED ON A RISING EDGE (FE)1CAPTURE EVENT 4 TRIGGERED ON A FALLING EDGE (FE)
5	CTRRST3	R/W	0h	COUNTER RESET ON CAPTURE EVENT 3:0DO NOT RESET COUNTER ON CAPTURE EVENT 3 (ABSOLUTE TIME STAMP)1RESET COUNTER AFTER EVENT 3 TIME-STAMP HAS BEEN CAPTURED(USED IN DIFFERENCE MODE OPERATION)
4	CAP3POL	R/W	0h	CAPTURE EVENT 3 POLARITY SELECT:0CAPTURE EVENT 3 TRIGGERED ON A RISING EDGE (FE)1CAPTURE EVENT 3 TRIGGERED ON A FALLING EDGE (FE)
3	CTRRST2	R/W	0h	COUNTER RESET ON CAPTURE EVENT 2:0DO NOT RESET COUNTER ON CAPTURE EVENT 2 (ABSOLUTE TIME STAMP)1RESET COUNTER AFTER EVENT 2 TIME-STAMP HAS BEEN CAPTURED(USED IN DIFFERENCE MODE OPERATION)
2	CAP2POL	R/W	0h	CAPTURE EVENT 2 POLARITY SELECT:0CAPTURE EVENT 2 TRIGGERED ON A RISING EDGE (FE)1CAPTURE EVENT 2 TRIGGERED ON A FALLING EDGE (FE)
1	CTRRST1	R/W	0h	COUNTER RESET ON CAPTURE EVENT 1:0DO NOT RESET COUNTER ON CAPTURE EVENT 1 (ABSOLUTE TIME STAMP)1RESET COUNTER AFTER EVENT 1 TIME-STAMP HAS BEEN CAPTURED(USED IN DIFFERENCE MODE OPERATION)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 4-218. ECCTL1 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	CAP1POL	R/W	0h	CAPTURE EVENT 1 POLARITY SELECT:0CAPTURE EVENT 1 TRIGGERED ON A RISING EDGE (FE)1CAPTURE EVENT 1 TRIGGERED ON A FALLING EDGE (FE)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.11.6 CNTPHS Register (offset = 4h) [reset = 0h]

COUNTER PHASE CONTROL REGISTER

CNTPHS is shown in Figure 4-213 and described in Table 4-219.

Figure 4-213. CNTPHS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNTPHS																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-219. CNTPHS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CNTPHS	R/W	0h	COUNTER PHASE VALUE REGISTER THAT CAN BE PROGRAMMED FOR PHASE LAG/LEAD. THIS REGISTER SHADOWS TSCNT AND IS LOADED INTO TSCNT UPON EITHER A SYNCI EVENT OR S/W FORCE VIA A CONTROL BIT.USED TO ACHIEVE PHASE CONTROL SYNC WITH RESPECT TO OTHER ECAP AND EPWM TIME-BASES.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.11.7 ECCTL2 Register (offset = 28h) [reset = 6h]

ECAP CONTROL REGISTER 2

ECCTL2 is shown in Figure 4-214 and described in Table 4-220.

Figure 4-214. ECCTL2 Register

15	14	13	12	11	10	9	8
FILTER					APWMPOL	CAP/APWM	SWSYNC
R-0h					R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
SYNCO_SEL		SYNCI_EN	TSCNTSTP	REARM/RESET	STOPVALUE		CONT/ONESHT
R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-3h		R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-220. ECCTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
15-11	FILTER	R	0h	
10	APWMPOL	R/W	0h	APWM OUTPUT POLARITY SELECT:0OUTPUT IS ACTIVE HIGH (I.E. COMPARE VALUE DEFINES HIGH TIME)1OUTPUT IS ACTIVE LOW (I.E. COMPARE VALUE DEFINES LOW TIME)NOTE: THIS IS APPLICABLE ONLY IN APWM OPERATING MODE
9	CAP/APWM	R/W	0h	CAP/APWM OPERATING MODE SELECT:0ECAP MODULE OPERATES IN CAPTURE MODETHIS MODE FORCES THE FOLLOWING CONFIGURATION:1) INHIBITS TSCNT RESETS VIA PRD_EQ EVENT2) INHIBITS SHADOW LOADS ON CAP1 and#38 AMP 2 REGISTERS3) PERMITS USER TO ENABLE CAP1-4 REGISTER LOAD4) CAPX/APWMX PIN OPERATES AS A CAPTURE INPUT*1ECAP MODULE OPERATES IN APWM MODETHIS MODE FORCES THE FOLLOWING CONFIGURATION:1) RESETS TSCNT ON PRD_EQ EVENT (PERIOD BOUNDARY)2) PERMITS SHADOW LOADING ON CAP1 and#38 AMP 2 REGISTERS3) DISABLES LOADING OF TIME-STAMPS INTO CAP1-4 REGS4) CAPX/APWMX PIN OPERATES AS A APWM OUTPUT*
8	SWSYNC	R/W	0h	SOFTWARE FORCED COUNTER (TSCNT) SYNC ING:0WRITING A ZERO HAS NO EFFECT WILL ALWAYS RETURN A ZERO1WRITING A ONE WILL FORCE A TSCNT SHADOW LOAD OF CURRENT ECAP MODULE AND ANY ECAP MODULES DOWN-STREAM PROVIDING THE SYNCO_SEL BITS ARE 0,0. AFTER WRITING A ONE THIS BIT RETURNS TO A ZERO.NOTE: THIS PROVIDES A CONVENIENT S/W METHOD TO SYNCHRONIZE SOME OR ALL ECAP TIMEBASES. IN APWM MODE THE SYNC ING CAN ALSO BE DONE VIA THE PRD_EQ EVENT.
7-6	SYNCO_SEL	R/W	0h	SYNC-OUT SELECT:0,0SELECT SYNC-IN EVENT TO BE THE SYNC-OUT SIGNAL (PASS THROUGH)0,1SELECT PRD_EQ EVENT TO BE THE SYNC-OUT SIGNAL1,0DISABLE SYNC OUT SIGNAL1,1DISABLE SYNC OUT SIGNALNOTE: SELECTION PRD_EQ IS MEANINGFUL ONLY IN APWM MODE, HOWEVER CAN STILL BE CHOSEN IN CAP MODE IF USER BELIEVES IT TO BE USEFUL.
5	SYNCI_EN	R/W	0h	COUNTER (TSCNT) SYNC-IN SELECT MODE:0DISABLE SYNC-IN OPTION1ENABLE COUNTER (TSCNT) TO BE LOADED FROM CNTPHS REGISTER UPON EITHER A SYNCI SIGNAL OR A S/W FORCE EVENT.
4	TSCNTSTP	R/W	0h	COUNTER STOP (FREEZE) CONTROL:0COUNTER STOPPED1COUNTER FREE RUNNING

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 4-220. ECCTL2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	REARM/RESET	R/W	0h	ONE-SHOT RE-ARMING , I.E. WAIT FOR STOP TRIGGER:WRITING A ONE ARMS THE ONE-SHOT SEQUENCE, I.E. 1) RESETS THE MOD4 COUNTER TO ZERO2) UN-FREEZES THE MOD4 COUNTER3) ENABLES CAPTURE REGISTER LOADSWRITING A ZERO HAS NO EFFECT. ALWAYS RETURNS A 0.NOTE: THE RE-ARM FUNCTION IS VALID IN ONESHT OR CONTINOUS MODE.
2-1	STOPVALUE	R/W	3h	STOP VALUE FOR ONE-SHOT MODE:THIS IS THE NUMBER (BETWEEN 1-4) OF CAPTURES ALLOWED TO OCCUR BEFORE THE CAP(1-4) REGISTERS ARE FROZEN, I.E.CAPTURE SEQUENCE IS STOPPED.0,0STOP AFTER CAPTURE EVENT 10,1STOP AFTER CAPTURE EVENT 21,0STOP AFTER CAPTURE EVENT 31,1STOP AFTER CAPTURE EVENT 4NOTES: [1] STOPVALUE IS COMPARED TO MOD4 COUNTER, WHEN EQUAL, 2 ACTIONS OCCUR:1) MOD4 COUNTER IS STOPPED (FROZEN)2) CAPTURE REGISTER LOADS ARE INHIBITED[2] IN ONE SHOT MODE, FURTHER INTERRUPT EVENTS ARE BLOCKED UNTIL WE RE-ARM, ONCE THE NUMBER OF EVENTS CAPTURED HAS BEEN REACHED.
0	CONT/ONESHT	R/W	0h	CONTINUOUS OR ONESHOT MODE CONTROL:(APPLICABLE ONLY IN CAPTURE MODE)0OPERATE IN CONTINUOUS MODE1OPERATE IN ONE-SHOT MODE

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.11.8 ECEINT Register (offset = 2Ah) [reset = 0h]

ECAP INTERRUPT ENABLE REGISTER

ECEINT is shown in Figure 4-215 and described in Table 4-221.

Figure 4-215. ECEINT Register

15	14	13	12	11	10	9	8
_RESV1							
R-0h							

7	6	5	4	3	2	1	0
CMPEQ	PRDEQ	CNTOVF	CEVT4	CEVT3	CEVT2	CEVT1	_RESV0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-221. ECEINT Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	_RESV1	R	0h	
7	CMPEQ	R/W	0h	COMPARE EQUAL INTERRUPT ENABLE: 0DISABLED COMPARE EQUAL AS AN INTERRUPT SOURCE1ENABLE COMPARE EQUAL AS AN INTERRUPT SOURCE
6	PRDEQ	R/W	0h	PERIOD EQUAL INTERRUPT ENABLE: 0DISABLED PERIOD EQUAL AS AN INTERRUPT SOURCE1ENABLE PERIOD EQUAL AS AN INTERRUPT SOURCE
5	CNTOVF	R/W	0h	COUNTER OVERFLOW INTERRUPT ENABLE: 0DISABLED COUNTER OVERFLOW AS AN INTERRUPT SOURCE1ENABLE COUNTER OVERFLOW AS AN INTERRUPT SOURCE
4	CEVT4	R/W	0h	CAPTURE EVENT 4 INTERRUPT ENABLE: 0DISABLED CAPTURE EVENT 1 AS AN INTERRUPT SOURCE1ENABLE CAPTURE EVENT 1 AS AN INTERRUPT SOURCE
3	CEVT3	R/W	0h	CAPTURE EVENT 3 INTERRUPT ENABLE: 0DISABLED CAPTURE EVENT 1 AS AN INTERRUPT SOURCE1ENABLE CAPTURE EVENT 1 AS AN INTERRUPT SOURCE
2	CEVT2	R/W	0h	CAPTURE EVENT 2 INTERRUPT ENABLE: 0DISABLED CAPTURE EVENT 1 AS AN INTERRUPT SOURCE1ENABLE CAPTURE EVENT 1 AS AN INTERRUPT SOURCE
1	CEVT1	R/W	0h	CAPTURE EVENT 1 INTERRUPT ENABLE: 0DISABLED CAPTURE EVENT 1 AS AN INTERRUPT SOURCE1ENABLE CAPTURE EVENT 1 AS AN INTERRUPT SOURCE
0	_RESV0	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.11.9 ECFLG Register (offset = 2Ch) [reset = 0h]

ECAP INTERRUPT FLAG REGISTER

ECFLG is shown in Figure 4-216 and described in Table 4-222.

Figure 4-216. ECFLG Register

15	14	13	12	11	10	9	8
_RESV0							
R-0h							

7	6	5	4	3	2	1	0
CMPEQ	PRDEQ	CNTOVF	CEVT4	CEVT3	CEVT2	CEVT1	INT
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-222. ECFLG Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	_RESV0	R	0h	
7	CMPEQ	R	0h	COMPARE EQUAL STATUS FLAG: READING A 1 ON THIS BIT INDICATES THE COUNTER (TSCNT) REACHED THE COMPARE REGISTER VALUE (ACMP)READING A 0 INDICATES NO EVENT OCCURREDNOTE: THIS FLAG IS ONLY ACTIVE IN APWM MODE.
6	PRDEQ	R	0h	PERIOD EQUAL STATUS FLAG: READING A 1 ON THIS BIT INDICATES THE COUNTER (TSCNT) REACHED THE PERIOD REGISTER VALUE (APER) AND WAS RESET.READING A 0 INDICATES NO EVENT OCCURREDNOTES: THIS FLAG IS ONLY ACTIVE IN APWM MODE.
5	CNTOVF	R	0h	COUNTER OVERFLOW STATUS FLAG: READING A 1 ON THIS BIT INDICATES THE COUNTER (TSCNT) HAS MADE THE TRANSITION FROM FFFFFFFF 00000000READING A 0 INDICATES NO EVENT OCCURRED.NOTE: THIS FLAG IS ACTIVE IN CAP and#38 AMP APWM MODE.
4	CEVT4	R	0h	CAPTURE EVENT 4 STATUS FLAG: READING A 1 ON THIS BIT INDICATES THE FOURTH EVENT OCCURRED AT ECAPX PIN.READING A 0 INDICATES NO EVENT OCCURRED.NOTE: THIS FLAG IS ONLY ACTIVE IN CAP MODE.
3	CEVT3	R	0h	CAPTURE EVENT 3 STATUS FLAG: READING A 1 ON THIS BIT INDICATES THE THIRD EVENT OCCURRED AT ECAPX PIN.READING A 0 INDICATES NO EVENT OCCURREDNOTE: THIS FLAG IS ONLY ACTIVE IN CAP MODE.
2	CEVT2	R	0h	CAPTURE EVENT 2 STATUS FLAG: READING A 1 ON THIS BIT INDICATES THE SECOND EVENT OCCURRED AT ECAPX PIN.READING A 0 INDICATES NO EVENT OCCURREDNOTE: THIS FLAG IS ONLY ACTIVE IN CAP MODE.
1	CEVT1	R	0h	CAPTURE EVENT 1 STATUS FLAG: READING A 1 ON THIS BIT INDICATES THE FIRST EVENT OCCURRED AT ECAPX PIN.READING A 0 INDICATES NO EVENT OCCURREDNOTE: THIS FLAG IS ONLY ACTIVE IN CAP MODE.
0	INT	R	0h	GLOBAL INTERRUPT STATUS FLAG: READING A 1 ON THIS BIT INDICATES THAT AN INTERRUPT WAS GENERATED FROM ONE OF THE FOLLOWING EVENTS.READING A 0 INDICATES NO INTERRUPT GENERATED.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.11.10 ECCLR Register (offset = 30h) [reset = 0h]

ECAP INTERRUPT CLEAR REGISTER

ECCLR is shown in Figure 4-217 and described in Table 4-223.

Figure 4-217. ECCLR Register

15	14	13	12	11	10	9	8
_RESV0							
R-0h							

7	6	5	4	3	2	1	0
CMPEQ	PRDEQ	CNTOVF	CEVT4	CEVT3	CEVT2	CEVT1	INT
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-223. ECCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	_RESV0	R	0h	
7	CMPEQ	R/W	0h	COMPARE EQUAL STATUS FLAG: WRITING A 1 WILL CLEAR THE CMPEQ FLAG CONDITION.WRITING A 0 WILL HAVE NO EFFECT.ALWAYS READS BACK A 0.
6	PRDEQ	R/W	0h	PERIOD EQUAL STATUS FLAG: WRITING A 1 WILL CLEAR THE PRDEQ FLAG CONDITION.WRITING A 0 WILL HAVE NO EFFECT.ALWAYS READS BACK A 0.
5	CNTOVF	R/W	0h	COUNTER OVERFLOW STATUS FLAG: WRITING A 1 WILL CLEAR THE CNTOVF FLAG CONDITION.WRITING A 0 WILL HAVE NO EFFECT.ALWAYS READS BACK A 0.
4	CEVT4	R/W	0h	CAPTURE EVENT 4 STATUS FLAG: WRITING A 1 WILL CLEAR THE CEVT3 FLAG CONDITION.WRITING A 0 WILL HAVE NO EFFECT.ALWAYS READS BACK A 0.
3	CEVT3	R/W	0h	CAPTURE EVENT 3 STATUS FLAG: WRITING A 1 WILL CLEAR THE CEVT3 FLAG CONDITION.WRITING A 0 WILL HAVE NO EFFECT.ALWAYS READS BACK A 0.
2	CEVT2	R/W	0h	CAPTURE EVENT 2 STATUS FLAG: WRITING A 1 WILL CLEAR THE CEVT2 FLAG CONDITION.WRITING A 0 WILL HAVE NO EFFECT.ALWAYS READS BACK A 0.
1	CEVT1	R/W	0h	CAPTURE EVENT 1 STATUS FLAG: WRITING A 1 WILL CLEAR THE CEVT1 FLAG CONDITION.WRITING A 0 WILL HAVE NO EFFECT.ALWAYS READS BACK A 0.
0	INT	R/W	0h	GLOBAL INTERRUPT CLEAR FLAG: WRITING A 1 WILL CLEAR THE INT FLAG AND ENABLE FURTHER INTERRUPTS TO BE GENERATED IF ANY OF THE EVENT FLAGS ARE SET TO 1.WRITING A 0 WILL HAVE NO EFFECT.ALWAYS READS BACK A 0.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.11.11 ECFRC Register (offset = 34h) [reset = 0h]

ECAP INTERRUPT FORCING REGISTER

ECFRC is shown in Figure 4-218 and described in Table 4-224.

Figure 4-218. ECFRC Register

15	14	13	12	11	10	9	8
_RESV1							
R-0h							

7	6	5	4	3	2	1	0
CMPEQ	PRDEQ	CNTOVF	CEVT4	CEVT3	CEVT2	CEVT1	_RESV0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-224. ECFRC Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	_RESV1	R	0h	
7	CMPEQ	R/W	0h	FORCE COMPARE EQUAL: WRITING A 1 TO THIS BIT WILL SET THE CMPEQ FLAG BIT.WRITING OF 0 WILL BE IGNORED. ALWAYS READS BACK A 0.
6	PRDEQ	R/W	0h	FORCE PERIOD EQUAL: WRITING A 1 TO THIS BIT WILL SET THE PRDEQ FLAG BIT.WRITING OF 0 WILL BE IGNORED. ALWAYS READS BACK A 0.
5	CNTOVF	R/W	0h	FORCE COUNTER OVERFLOW: WRITING A 1 TO THIS BIT WILL SET THE CNTOVF FLAG BIT.WRITING OF 0 WILL BE IGNORED. ALWAYS READS BACK A 0.
4	CEVT4	R/W	0h	FORCE CAPTURE EVENT 4: WRITING A 1 TO THIS BIT WILL SET THE CEVT4 FLAG BIT.WRITING OF 0 WILL BE IGNORED. ALWAYS READS BACK A 0.
3	CEVT3	R/W	0h	FORCE CAPTURE EVENT 3: WRITING A 1 TO THIS BIT WILL SET THE CEVT3 FLAG BIT.WRITING OF 0 WILL BE IGNORED. ALWAYS READS BACK A 0.
2	CEVT2	R/W	0h	FORCE CAPTURE EVENT 2: WRITING A 1 TO THIS BIT WILL SET THE CEVT2 FLAG BIT.WRITING OF 0 WILL BE IGNORED. ALWAYS READS BACK A 0.
1	CEVT1	R/W	0h	FORCE CAPTURE EVENT 1: WRITING A 1 TO THIS BIT WILL SET THE CEVT1 FLAG BIT.WRITING OF 0 WILL BE IGNORED. ALWAYS READS BACK A 0.
0	_RESV0	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.11.12 CAP1 Register (offset = 8h) [reset = 0h]

CAPTURE-1 REGISTER

CAP1 is shown in Figure 4-219 and described in Table 4-225.

Figure 4-219. CAP1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAP1																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-225. CAP1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAP1	R/W	0h	THIS REGISTER CAN BE LOADED (WRITTEN) BY :TIME-STAMP (I.E. COUNTER VALUE) DURING A CAPTURE EVENTS/W MAY BE USEFUL FOR TEST PURPOSES / INITIALISATIONAPRD SHADOW REGISTER (I.E. CAP3) WHEN USED IN APWM MODE

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.11.13 PID Register (offset = 5Ch) [reset = 44D22100h]

ECAP PERIPHERAL ID REGISTER

PID is shown in Figure 4-220 and described in Table 4-226.

Figure 4-220. PID Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVID																															
R-44D22100h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-226. PID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REVID	R	44D22100h	

4.12 PRUSS_MII_RT_CFG Registers

Table 4-227 lists the memory-mapped registers for the PRUSS_MII_RT_CFG. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 4-227 should be considered as reserved locations and the register contents should not be modified.

Table 4-227. PRUSS_MII_RT_CFG REGISTERS

Offset	Acronym	Register Description	Section
0h	RXCFG0	MII RXCFG 0 REGISTER	Section 4.12.1
10h	TXCFG0	MII TXCFG 0 REGISTER	Section 4.12.2
14h	TXCFG1	MII TXCFG 1 REGISTER	Section 4.12.3
20h	TX_CRC0	MII TXCRC 0 REGISTER	Section 4.12.4
24h	TX_CRC1	MII TXCRC 1 REGISTER	Section 4.12.5
4h	RXCFG1	MII RXCFG 1 REGISTER	Section 4.12.6
30h	TX_IPG0	MII TXIPG 0 REGISTER	Section 4.12.7
34h	TX_IPG1	MII TXIPG 1 REGISTER	Section 4.12.8
38h	PRS0	MII PORT STATUS 0 REGISTER	Section 4.12.9
3Ch	PRS1	MII PORT STATUS 1 REGISTER	Section 4.12.10
40h	RX_FRMS0	MII RXFRMS 0 REGISTER	Section 4.12.11
44h	RX_FRMS1	MII RXFRMS 1 REGISTER	Section 4.12.12
48h	RX_PCNT0	MII RXPCNT 0 REGISTER	Section 4.12.13
4Ch	RX_PCNT1	MII RXPCNT 1 REGISTER	Section 4.12.14
50h	RX_ERR0	MII RXERR 0 REGISTER	Section 4.12.15
54h	RX_ERR1	MII RXERR 1 REGISTER	Section 4.12.16

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.12.1 RXCFG0 Register (offset = 0h) [reset = 0h]

MII RXCFG 0 REGISTER

RXCFG0 is shown in Figure 4-221 and described in Table 4-228.

Figure 4-221. RXCFG0 Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved	RX_AUTO_FWD_PRE	RX_BYTE_SWAP	RX_L2_EN	RX_MUX_SEL	RX_CUT_PREAMBLE	Reserved	RX_ENABLE
	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h		R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-228. RXCFG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
6	RX_AUTO_FWD_PRE	R/W	0h	
5	RX_BYTE_SWAP	R/W	0h	
4	RX_L2_EN	R/W	0h	
3	RX_MUX_SEL	R/W	0h	
2	RX_CUT_PREAMBLE	R/W	0h	
0	RX_ENABLE	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.12.2 TXCFG0 Register (offset = 10h) [reset = 400100h]

MII TXCFG 0 REGISTER

TXCFG0 is shown in Figure 4-222 and described in Table 4-229.

Figure 4-222. TXCFG0 Register

31	30	29	28	27	26	25	24
Reserved	TX_CLK_DELAY			Reserved		TX_START_DELAY	
	R/W-0h					R/W-40h	

23	22	21	20	19	18	17	16
TX_START_DELAY							
R/W-40h							

15	14	13	12	11	10	9	8
Reserved					PRE_TX_AUTO_ESC_ERR	PRE_TX_AUTO_SEQUENCE	TX_MUX_SEL
					R/W-0h	R/W-0h	R/W-1h

7	6	5	4	3	2	1	0
Reserved				TX_BYTE_SWAP	TX_EN_MODE	TX_AUTO_PREAMBLE	TX_ENABLE
				R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-229. TXCFG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
30-28	TX_CLK_DELAY	R/W	0h	
25-16	TX_START_DELAY	R/W	40h	
10	PRE_TX_AUTO_ESC_ERR	R/W	0h	
9	PRE_TX_AUTO_SEQUENCE	R/W	0h	
8	TX_MUX_SEL	R/W	1h	
3	TX_BYTE_SWAP	R/W	0h	
2	TX_EN_MODE	R/W	0h	
1	TX_AUTO_PREAMBLE	R/W	0h	
0	TX_ENABLE	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.12.3 TXCFG1 Register (offset = 14h) [reset = 400000h]

MII TXCFG 1 REGISTER

TXCFG1 is shown in Figure 4-223 and described in Table 4-230.

Figure 4-223. TXCFG1 Register

31	30	29	28	27	26	25	24
Reserved	TX_CLK_DELAY			Reserved		TX_START_DELAY	
	R/W-0h					R/W-40h	

23	22	21	20	19	18	17	16
TX_START_DELAY							
R/W-40h							

15	14	13	12	11	10	9	8
Reserved					PRE_TX_AUTO_ESC_ERR	PRE_TX_AUTO_SEQUENCE	TX_MUX_SEL
					R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
Reserved				TX_BYTE_SWAP	TX_EN_MODE	TX_AUTO_PREAMBLE	TX_ENABLE
				R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-230. TXCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
30-28	TX_CLK_DELAY	R/W	0h	
25-16	TX_START_DELAY	R/W	40h	
10	PRE_TX_AUTO_ESC_ERR	R/W	0h	
9	PRE_TX_AUTO_SEQUENCE	R/W	0h	
8	TX_MUX_SEL	R/W	0h	
3	TX_BYTE_SWAP	R/W	0h	
2	TX_EN_MODE	R/W	0h	
1	TX_AUTO_PREAMBLE	R/W	0h	
0	TX_ENABLE	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.12.4 TX_CRC0 Register (offset = 20h) [reset = 0h]

MII TXCRC 0 REGISTER

TX_CRC0 is shown in Figure 4-224 and described in Table 4-231.

Figure 4-224. TX_CRC0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_CRC																															
R-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-231. TX_CRC0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_CRC	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.12.5 TX_CRC1 Register (offset = 24h) [reset = 0h]

MII TXCRC 1 REGISTER

TX_CRC1 is shown in Figure 4-225 and described in Table 4-232.

Figure 4-225. TX_CRC1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_CRC																															
R-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-232. TX_CRC1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_CRC	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.12.6 RXCFG1 Register (offset = 4h) [reset = 8h]

MII RXCFG 1 REGISTER

RXCFG1 is shown in Figure 4-226 and described in Table 4-233.

Figure 4-226. RXCFG1 Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved	RX_AUTO_FWD_PRE	RX_BYTE_SWAP	RX_L2_EN	RX_MUX_SEL	RX_CUT_PREAMBLE	Reserved	RX_ENABLE
	R/W-0h	R/W-0h	R/W-0h	R/W-1h	R/W-0h		R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-233. RXCFG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
6	RX_AUTO_FWD_PRE	R/W	0h	
5	RX_BYTE_SWAP	R/W	0h	
4	RX_L2_EN	R/W	0h	
3	RX_MUX_SEL	R/W	1h	
2	RX_CUT_PREAMBLE	R/W	0h	
0	RX_ENABLE	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.12.7 TX_IPG0 Register (offset = 30h) [reset = 28h]

MII TXIPG 0 REGISTER

TX_IPG0 is shown in Figure 4-227 and described in Table 4-234.

Figure 4-227. TX_IPG0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																						TX_IPG									
																						R/W-28h									

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-234. TX_IPG0 Register Field Descriptions

Bit	Field	Type	Reset	Description
9-0	TX_IPG	R/W	28h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.12.8 TX_IPG1 Register (offset = 34h) [reset = 28h]

MII TXIPG 1 REGISTER

TX_IPG1 is shown in Figure 4-228 and described in Table 4-235.

Figure 4-228. TX_IPG1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																						TX_IPG									
																						R/W-28h									

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-235. TX_IPG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
9-0	TX_IPG	R/W	28h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.12.9 PRS0 Register (offset = 38h) [reset = 0h]

MII PORT STATUS 0 REGISTER

PRS0 is shown in Figure 4-229 and described in Table 4-236.

Figure 4-229. PRS0 Register

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						MII_CRS	MII_COL
						R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-236. PRS0 Register Field Descriptions

Bit	Field	Type	Reset	Description
1	MII_CRS	R	0h	
0	MII_COL	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.12.10 PRS1 Register (offset = 3Ch) [reset = 0h]

MII PORT STATUS 1 REGISTER

PRS1 is shown in Figure 4-230 and described in Table 4-237.

Figure 4-230. PRS1 Register

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						MII_CRS	MII_COL
						R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-237. PRS1 Register Field Descriptions

Bit	Field	Type	Reset	Description
1	MII_CRS	R	0h	
0	MII_COL	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.12.11 RX_FRMS0 Register (offset = 40h) [reset = 5F1003Fh]

MII RXFRMS 0 REGISTER

RX_FRMS0 is shown in Figure 4-231 and described in Table 4-238.

Figure 4-231. RX_FRMS0 Register

31	30	29	28	27	26	25	24
RX_MAX_FRM							
R/W-5F1h							

23	22	21	20	19	18	17	16
RX_MAX_FRM							
R/W-5F1h							

15	14	13	12	11	10	9	8
RX_MIN_FRM							
R/W-3Fh							

7	6	5	4	3	2	1	0
RX_MIN_FRM							
R/W-3Fh							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-238. RX_FRMS0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RX_MAX_FRM	R/W	5F1h	
15-0	RX_MIN_FRM	R/W	3Fh	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.12.12 RX_FRMS1 Register (offset = 44h) [reset = 5F1003Fh]

MII RXFRMS 1 REGISTER

RX_FRMS1 is shown in Figure 4-232 and described in Table 4-239.

Figure 4-232. RX_FRMS1 Register

31	30	29	28	27	26	25	24
RX_MAX_FRM							
R/W-5F1h							

23	22	21	20	19	18	17	16
RX_MAX_FRM							
R/W-5F1h							

15	14	13	12	11	10	9	8
RX_MIN_FRM							
R/W-3Fh							

7	6	5	4	3	2	1	0
RX_MIN_FRM							
R/W-3Fh							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-239. RX_FRMS1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RX_MAX_FRM	R/W	5F1h	
15-0	RX_MIN_FRM	R/W	3Fh	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.12.13 RX_PCNT0 Register (offset = 48h) [reset = E1h]

MII RXPCNT 0 REGISTER

RX_PCNT0 is shown in Figure 4-233 and described in Table 4-240.

Figure 4-233. RX_PCNT0 Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
RX_MAX_PCNT				RX_MIN_PCNT			
R/W-Eh				R/W-1h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-240. RX_PCNT0 Register Field Descriptions

Bit	Field	Type	Reset	Description
7-4	RX_MAX_PCNT	R/W	Eh	
3-0	RX_MIN_PCNT	R/W	1h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.12.14 RX_PCNT1 Register (offset = 4Ch) [reset = E1h]

MII RXPCNT 1 REGISTER

RX_PCNT1 is shown in Figure 4-234 and described in Table 4-241.

Figure 4-234. RX_PCNT1 Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
RX_MAX_PCNT				RX_MIN_PCNT			
R/W-Eh				R/W-1h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-241. RX_PCNT1 Register Field Descriptions

Bit	Field	Type	Reset	Description
7-4	RX_MAX_PCNT	R/W	Eh	
3-0	RX_MIN_PCNT	R/W	1h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.12.15 RX_ERR0 Register (offset = 50h) [reset = 0h]

MII RXERR 0 REGISTER

RX_ERR0 is shown in Figure 4-235 and described in Table 4-242.

Figure 4-235. RX_ERR0 Register

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				RX_MAX_FRM_ERR	RX_MIN_FRM_ERR	RX_MAX_PCNT_ERR	RX_MIN_PCNT_ERR
				R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-242. RX_ERR0 Register Field Descriptions

Bit	Field	Type	Reset	Description
3	RX_MAX_FRM_ERR	R/W	0h	
2	RX_MIN_FRM_ERR	R/W	0h	
1	RX_MAX_PCNT_ERR	R/W	0h	
0	RX_MIN_PCNT_ERR	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.12.16 RX_ERR1 Register (offset = 54h) [reset = 0h]

MII RXERR 1 REGISTER

RX_ERR1 is shown in Figure 4-236 and described in Table 4-243.

Figure 4-236. RX_ERR1 Register

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				RX_MAX_FRM_ERR	RX_MIN_FRM_ERR	RX_MAX_PCNT_ERR	RX_MIN_PCNT_ERR
				R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-243. RX_ERR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
3	RX_MAX_FRM_ERR	R/W	0h	
2	RX_MIN_FRM_ERR	R/W	0h	
1	RX_MAX_PCNT_ERR	R/W	0h	
0	RX_MIN_PCNT_ERR	R/W	0h	

4.13 PRUSS_MII_MDIO Registers

Table 4-244 lists the memory-mapped registers for the PRUSS_MII_MDIO. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 4-244 should be considered as reserved locations and the register contents should not be modified.

Table 4-244. PRUSS_MII_MDIO REGISTERS

Offset	Acronym	Register Description	Section
0h	VER	MODULE VERSION REGISTER	Section 4.13.1
Ch	LINK	PHY LINK STATUS REGISTER	Section 4.13.2
80h	USERACCESS0	USER ACCESS REGISTER0	Section 4.13.3
84h	USERPHYSEL0	USER PHY SELECT REGISTER0	Section 4.13.4
88h	USERACCESS1	USER ACCESS REGISTER1	Section 4.13.5
8Ch	USERPHYSEL1	USER PHY SELECT REGISTER1	Section 4.13.6
10h	LINKINTRAW	LINK STATUS CHANGE INTERRUPT REGISTER (RAW VALUE)	Section 4.13.7
14h	LINKINTMASKED	LINK STATUS CHANGE INTERRUPT REGISTER (MASKED VALUE)	Section 4.13.8
20h	USERINTRAW	USER COMMAND COMPLETE INTERRUPT REGISTER (RAW VALUE)	Section 4.13.9
24h	USERINTMASKED	USER COMMAND COMPLETE INTERRUPT REGISTER (MASKED VALUE)	Section 4.13.10
4h	CONTROL	MODULE CONTROL REGISTER	Section 13.11.30
28h	USERINTMASKSET	USER INTERRUPT MASK SET REGISTER	Section 4.13.12
2Ch	USERINTMASKCLR	USER INTERRUPT MASK CLEAR REGISTER	Section 4.13.13
8h	ALIVE	PHY ACKNOWLEDGE STATUS REGISTER	Section 4.13.14

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.13.1 VER Register (offset = 0h) [reset = 70105h]

MODULE VERSION REGISTER

VER is shown in Figure 4-237 and described in Table 4-245.

Figure 4-237. VER Register

31	30	29	28	27	26	25	24
MODID							
R-7h							

23	22	21	20	19	18	17	16
MODID							
R-7h							

15	14	13	12	11	10	9	8
REVMAJ							
R-1h							

7	6	5	4	3	2	1	0
REVMIN							
R-5h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-245. VER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	MODID	R	7h	
15-8	REVMAJ	R	1h	
7-0	REVMIN	R	5h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.13.2 LINK Register (offset = Ch) [reset = 0h]

PHY LINK STATUS REGISTER

LINK is shown in Figure 4-238 and described in Table 4-246.

Figure 4-238. LINK Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LINK																															
R-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-246. LINK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LINK	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.13.3 USERACCESS0 Register (offset = 80h) [reset = 0h]

USER ACCESS REGISTER0

USERACCESS0 is shown in Figure 4-239 and described in Table 4-247.

Figure 4-239. USERACCESS0 Register

31	30	29	28	27	26	25	24
GO	WRITE	ACK	Reserved			REGADR	
R/W-0h	R/W-0h	R/W-0h				R/W-0h	

23	22	21	20	19	18	17	16
REGADR			PHYADR				
R/W-0h			R/W-0h				

15	14	13	12	11	10	9	8
DATA							
R/W-0h							

7	6	5	4	3	2	1	0
DATA							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-247. USERACCESS0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GO	R/W	0h	
30	WRITE	R/W	0h	
29	ACK	R/W	0h	
25-21	REGADR	R/W	0h	
20-16	PHYADR	R/W	0h	
15-0	DATA	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.13.4 USERPHYSEL0 Register (offset = 84h) [reset = 0h]

USER PHY SELECT REGISTER0

USERPHYSEL0 is shown in Figure 4-240 and described in Table 4-248.

Figure 4-240. USERPHYSEL0 Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
LINKSEL	LINKINT_ENABLE	Reserved	PHYADR_MON				
R/W-0h	R/W-0h				R/W-0h		

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-248. USERPHYSEL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
7	LINKSEL	R/W	0h	
6	LINKINT_ENABLE	R/W	0h	
4-0	PHYADR_MON	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.13.5 USERACCESS1 Register (offset = 88h) [reset = 0h]

USER ACCESS REGISTER1

USERACCESS1 is shown in Figure 4-241 and described in Table 4-249.

Figure 4-241. USERACCESS1 Register

31	30	29	28	27	26	25	24
GO	WRITE	ACK	Reserved			REGADR	
R/W-0h	R/W-0h	R/W-0h				R/W-0h	

23	22	21	20	19	18	17	16
REGADR			PHYADR				
R/W-0h			R/W-0h				

15	14	13	12	11	10	9	8
DATA							
R/W-0h							

7	6	5	4	3	2	1	0
DATA							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-249. USERACCESS1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	GO	R/W	0h	
30	WRITE	R/W	0h	
29	ACK	R/W	0h	
25-21	REGADR	R/W	0h	
20-16	PHYADR	R/W	0h	
15-0	DATA	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.13.6 USERPHYSEL1 Register (offset = 8Ch) [reset = 0h]

USER PHY SELECT REGISTER1

USERPHYSEL1 is shown in Figure 4-242 and described in Table 4-250.

Figure 4-242. USERPHYSEL1 Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
LINKSEL	LINKINT_ENABLE	Reserved	PHYADR_MON				
R/W-0h	R/W-0h				R/W-0h		

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-250. USERPHYSEL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
7	LINKSEL	R/W	0h	
6	LINKINT_ENABLE	R/W	0h	
4-0	PHYADR_MON	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.13.7 LINKINTRAW Register (offset = 10h) [reset = 0h]

LINK STATUS CHANGE INTERRUPT REGISTER (RAW VALUE)

LINKINTRAW is shown in Figure 4-243 and described in Table 4-251.

Figure 4-243. LINKINTRAW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																														LINKINTRAW	
																														R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-251. LINKINTRAW Register Field Descriptions

Bit	Field	Type	Reset	Description
1-0	LINKINTRAW	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.13.8 LINKINTMASKED Register (offset = 14h) [reset = 0h]

LINK STATUS CHANGE INTERRUPT REGISTER (MASKED VALUE)

LINKINTMASKED is shown in Figure 4-244 and described in Table 4-252.

Figure 4-244. LINKINTMASKED Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																														LINKINTMASKED	
																														R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-252. LINKINTMASKED Register Field Descriptions

Bit	Field	Type	Reset	Description
1-0	LINKINTMASKED	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.13.9 USERINTRAW Register (offset = 20h) [reset = 0h]

USER COMMAND COMPLETE INTERRUPT REGISTER (RAW VALUE)

USERINTRAW is shown in Figure 4-245 and described in Table 4-253.

Figure 4-245. USERINTRAW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																														USERINT RAW	
																														-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-253. USERINTRAW Register Field Descriptions

Bit	Field	Type	Reset	Description
1-0	USERINTRAW		0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.13.10 USERINTMASKED Register (offset = 24h) [reset = 0h]

USER COMMAND COMPLETE INTERRUPT REGISTER (MASKED VALUE)

USERINTMASKED is shown in Figure 4-246 and described in Table 4-254.

Figure 4-246. USERINTMASKED Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																														USERINT MASKED	
																														-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-254. USERINTMASKED Register Field Descriptions

Bit	Field	Type	Reset	Description
1-0	USERINTMASKED		0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.13.11 CONTROL Register (offset = 4h) [reset = 810000FFh]

MODULE CONTROL REGISTER

CONTROL is shown in Figure 13-227 and described in Table 13-240.

Figure 4-247. CONTROL Register

31	30	29	28	27	26	25	24
IDLE	ENABLE	Reserved	HIGHEST_USER_CHANNEL				
R-1h	R/W-0h		R-1h				

23	22	21	20	19	18	17	16
Reserved			PREAMBLE	FAULT	FAULT_DETECT_ENABLE	INT_TEST_ENABLE	Reserved
			R/W-0h	-0h	R/W-0h	R/W-0h	

15	14	13	12	11	10	9	8
CLKDIV							
R/W-FFh							

7	6	5	4	3	2	1	0
CLKDIV							
R/W-FFh							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-255. CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IDLE	R	1h	
30	ENABLE	R/W	0h	
28-24	HIGHEST_USER_CHANNEL	R	1h	
20	PREAMBLE	R/W	0h	
19	FAULT		0h	
18	FAULT_DETECT_ENABLE	R/W	0h	
17	INT_TEST_ENABLE	R/W	0h	
15-0	CLKDIV	R/W	FFh	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.13.12 USERINTMASKSET Register (offset = 28h) [reset = 0h]

USER INTERRUPT MASK SET REGISTER

USERINTMASKSET is shown in Figure 4-248 and described in Table 4-256.

Figure 4-248. USERINTMASKSET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																														USERINT MASKED SET	
																														R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-256. USERINTMASKSET Register Field Descriptions

Bit	Field	Type	Reset	Description
1-0	USERINTMASKEDSET	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.13.13 USERINTMASKCLR Register (offset = 2Ch) [reset = 0h]

USER INTERRUPT MASK CLEAR REGISTER

USERINTMASKCLR is shown in Figure 4-249 and described in Table 4-257.

Figure 4-249. USERINTMASKCLR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																														USERINT MASKED CLR	
																														R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 4-257. USERINTMASKCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
1-0	USERINTMASKEDCLR	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4.13.14 ALIVE Register (offset = 8h) [reset = 0h]

PHY ACKNOWLEDGE STATUS REGISTER

ALIVE is shown in Figure 4-250 and described in Table 4-258.

Figure 4-250. ALIVE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALIVE																															
-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 4-258. ALIVE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ALIVE		0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Chapter 5

SPRUH73–July 2011

Graphics Accelerator (SGX)

This chapter describes the graphics accelerator for the device.

Topic **Page**

5.1 Introduction **598**
5.2 Integration **598**
5.3 Functional Description **598**
5.4 Use Cases **598**
5.5 Registers **599**

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.1 Introduction

5.2 Integration

5.3 Functional Description

5.4 Use Cases

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5 Registers

5.5.1 SGX530_MMAV10GS70 Registers

Table 5-1 lists the memory-mapped registers for the SGX530_MMAV10GS70. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 5-1 should be considered as reserved locations and the register contents should not be modified.

Table 5-1. SGX530_MMAV10GS70 REGISTERS

Offset	Acronym	Register Description	Section
0h	EUR_CR_CLKGATECTL	Core module clock gating controls: allows clocks to be forced off, forced on or operate under automatic pipeline activity based clock gating.	Section 5.5.1.1
64h	EUR_CR_PERF_COUNTER8	Performance counter status register	Section 5.5.1.2
400h	EUR_CR_PIXELBE	The PixelBE modules format the pixel data before writing to memory, this register contains global control signals.	Section 5.5.1.3
404h	EUR_CR_ISP_RENDER	The Image Synthesis Processor can be configured to operate in 3 different render modes	Section 5.5.1.4
408h	EUR_CR_ISP_RGN_BASE	This register defines the base address in memory of the Region Header reads by the ISP. Region headers are the first part of the display list and contain an entry per tile with information on global setup and a link address to parameters.	Section 5.5.1.5
40Ch	EUR_CR_ISP_RENDBOX1	When the render type is set to Fast 2D Render or Fast Scale Render, no region headers are read from memory. Instead, region headers are internally generated for all tiles enclosed in the bounding box defined by this register and the EUR_CR_ISP_RENDBOX2 register.	Section 5.5.1.6
410h	EUR_CR_ISP_RENDBOX2	When the render type is set to Fast 2D Render or Fast Scale Render, no region headers are read from memory. Instead, region headers are internally generated for all tiles enclosed in the bounding box defined by this register and the EUR_CR_ISP_RENDBOX1 register.	Section 5.5.1.7
414h	EUR_CR_ISP_IPFMISC	Global control register for the ISP parameter fetch.	Section 5.5.1.8
418h	EUR_CR_ISP_OGL_MODE	Global control register which modifies the pixel sample position to comply with OGL requirements.	Section 5.5.1.9
41Ch	EUR_CR_ISP_PERPENDICULAR		Section 5.5.1.10
420h	EUR_CR_ISP_CULLVALUE		Section 5.5.1.11
424h	EUR_CR_ISP_DBIAS		Section 5.5.1.12
428h	EUR_CR_ISP_START_RENDER	Any write to this register starts a 3D Render.	Section 5.5.1.13
42Ch	EUR_CR_3D_AA_MODE	Indicates to the 3D pipeline whether anti-aliasing is enabled or disabled	Section 5.5.1.14
430h	EUR_CR_ISP_BREAK	Object breakpoints can be inserted by the driver to interrupt a render part way and then resume the render from the point of interrupt at a later time, this register controls what happens when a breakpoint has been encountered.	Section 5.5.1.15
434h	EUR_CR_ISP_3DCONTEXT		Section 5.5.1.16
438h	EUR_CR_ISP_FPU		Section 5.5.1.17
43Ch	EUR_CR_TSP_PARAMETER_CACHE		Section 5.5.1.18
440h	EUR_CR_TSP_PARAMETER_FETCH		Section 5.5.1.19
480h	EUR_CR_ISP_ZLSCTL	Image Synthesis Processor Z Load/Store and format global control register	Section 5.5.1.20
484h	EUR_CR_ISP_ZLOAD_BASE	This register defines the base address in memory of the Z Buffer base address to load into the ISP for non-compressed ZLS formats.	Section 5.5.1.21

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 5-1. SGX530_MMAV10GS70 REGISTERS (continued)

Offset	Acronym	Register Description	Section
488h	EUR_CR_ISP_ZSTORE_BASE	This register defines the base address in memory of the Z Buffer base address to store data from the ISP for non-compressed ZLS formats.	Section 5.5.1.22
48Ch	EUR_CR_ISP_STENCIL_LOAD_BASE	This register defines the base address in memory of the Stencil Buffer base address to load into the ISP for Int24 and IEEE ZLS formats. This alternate stencil buffer base address is selectable based on the enable bit.	Section 5.5.1.23
490h	EUR_CR_ISP_STENCIL_STORE_BASE	This register defines the base address in memory of the Stencil Buffer base address to store data from the ISP for Int24 and IEEE ZLS formats. This alternate stencil buffer base address is selectable based on the enable bit.	Section 5.5.1.24
494h	EUR_CR_ISP_VISTEST_CTRL	This register can be used to reset the results of the 8 visibility test results. Write a 1 to the required bit field to clear the result register	Section 5.5.1.25
498h	EUR_CR_ISP_VISTEST_VISIBLE0		Section 5.5.1.26
49Ch	EUR_CR_ISP_VISTEST_VISIBLE1		Section 5.5.1.27
4A0h	EUR_CR_ISP_VISTEST_VISIBLE2		Section 5.5.1.28
4A4h	EUR_CR_ISP_VISTEST_VISIBLE3		Section 5.5.1.29
4A8h	EUR_CR_ISP_VISTEST_VISIBLE4		Section 5.5.1.30
4ACh	EUR_CR_ISP_VISTEST_VISIBLE5		Section 5.5.1.31
4B0h	EUR_CR_ISP_VISTEST_VISIBLE6		Section 5.5.1.32
4B4h	EUR_CR_ISP_VISTEST_VISIBLE7		Section 5.5.1.33
4B8h	EUR_CR_ISP_BGOBJDEPTH	The ISP operates by comparing depth values of incoming objects with the results of previous depth compares, in order to make sure there are no un-initialised values at the start of the tile or to cover pixels where there are no objects in the scene a default background object is configured under register control. This register provides the floating point depth value for this background object.	Section 5.5.1.34
4BCh	EUR_CR_ISP_BGOBJ	The ISP operates by comparing depth values of incoming objects with the results of previous depth compares, in order to make sure there is no un-initialised values at the start of the tile or to cover pixels where there are no objects in the scene a default background object is configured under register control. This register provides tag, mask and stencil information for this background object.	Section 5.5.1.35
4C4h	EUR_CR_ISP_BGOBJTAG	The ISP operates by comparing depth values of incoming objects with the results of previous depth compares, in order to make sure there are no unitialised values at the start of the tile or to cover pixels where there are no objects in the scene a default background object is configured under register control. This register provides TSP and Vertex information for this background object.	Section 5.5.1.36
4C8h	EUR_CR_ISP_MULTISAMPLECTL	This applies when anti-aliasing is enabled. This is used by the ISP to adjust the sample positions.	Section 5.5.1.37
4CCh	EUR_CR_ISP_SIG1		Section 5.5.1.38
4D0h	EUR_CR_ISP_SIG2		Section 5.5.1.39
4D4h	EUR_CR_ISP_SIG3		Section 5.5.1.40

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 5-1. SGX530_MMAV10GS70 REGISTERS (continued)

Offset	Acronym	Register Description	Section
4D8h	EUR_CR_ISP_SIG4		Section 5.5.1.41
4DCh	EUR_CR_ISP_TAGCTRL		Section 5.5.1.42
4E0h	EUR_CR_ISP_STATUS1		Section 5.5.1.43
4E4h	EUR_CR_ISP_STATUS2		Section 5.5.1.44
4E8h	EUR_CR_ISP_DEPTHSORT		Section 5.5.1.45
4ECh	EUR_CR_PIXELBE_EMIT	This register indicates if the PixelBE would stall the USE if there were another emit. '1' indicates that the PixelBE would stall the USE. '0' indicates that the PixelBE would not stall the USE.	Section 5.5.1.46
80h	EUR_CR_SOFT_RESET	Soft reset control register which drives all modules except the register bank. Write a '1' to reset and a '0' to clear	Section 5.5.1.47
600h	EUR_CR_DPM_3D_PAGE_TABLE_BASE	Effective on load 3D context, this register defines the base address of the page table being referenced in the process of de-allocating pages during a 3D render.	Section 5.5.1.48
604h	EUR_CR_DPM_3D_FREE_LIST	Between the FREE_LIST_HEAD and FREE_LIST_TAIL there must be a valid chain of linked pages. The linked list is stored in memory, at the EUR_CR_DPM_3D_PAGE_TABLE_BASE_ADDRESS and is effective on load 3D context.	Section 5.5.1.49
614h	EUR_CR_DPM_PDS_PAGE_THRESHOLD	This is the threshold in 4KB pages at which the hardware will stop double buffering the internal vertex processing buffers (and drop to single buffered mode). This is required in order to guarantee that a 3D render can run when running in dynamic parameter management mode. It should be set to 0.25MB less (in pages 0x40) than the ZLS threshold (or 0 in very small memory systems).	Section 5.5.1.50
618h	EUR_CR_DPM_TA_ALLOC_PAGE_TABLE_BASE	Effective on Loading the TA Page Table Context, this register defines the base address of the page table being referenced in the process of allocating pages by the TE and MTE.	Section 5.5.1.51
61Ch	EUR_CR_DPM_TA_ALLOC_FREE_LIST	Between the FREE_LIST_HEAD and FREE_LIST_TAIL there must be a valid chain of linked pages. The linked list is stored in memory, at the EUR_CR_DPM_TA_ALLOC_PAGE_TABLE_BASE_ADDRESS and is effective on Loading the TA Page Table Context.	Section 5.5.1.52
620h	EUR_CR_DPM_TA_PAGE_THRESHOLD		Section 5.5.1.53
624h	EUR_CR_DPM_ZLS_PAGE_THRESHOLD		Section 5.5.1.54
628h	EUR_CR_DPM_TA_GLOBAL_LIST		Section 5.5.1.55
62Ch	EUR_CR_DPM_STATE_TABLE_BASE	Effective immediately, this register defines the base address of the state table when it is read and written to/from memory. The state table itself contains the head, tail and number of pages allocated to every macrotile, and the global list. Each context needs a separate state page table.	Section 5.5.1.56
630h	EUR_CR_DPM_STATE_CONTEXT_ID		Section 5.5.1.57
634h	EUR_CR_DPM_CONTROL_TABLE_BASE	Effective immediately, this register defines the base address of the control table when it is read and written to/from memory. The control table itself contains the current offset into the page, and whether a page has been allocated, for each macrotile. Each context needs a separate state page table.	Section 5.5.1.58
638h	EUR_CR_DPM_START_OF_CONTEXT_PAGES_ALLOCATED		Section 5.5.1.59

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 5-1. SGX530_MMAV10GS70 REGISTERS (continued)

Offset	Acronym	Register Description	Section
63Ch	EUR_CR_DPM_3D_DEALLOCATE		Section 5.5.1.60
10h	EUR_CR_CORE_ID	Core ID Register	Section 5.5.1.61
640h	EUR_CR_DPM_ALLOC	A write to this register invalidates the current contents and causes a new allocation cycle	Section 5.5.1.62
644h	EUR_CR_DPM_DALLOC	A write to this register invalidates the current contents and causes a new de-allocation cycle	Section 5.5.1.63
648h	EUR_CR_DPM_TA_ALLOC		Section 5.5.1.64
64Ch	EUR_CR_DPM_3D		Section 5.5.1.65
658h	EUR_CR_DPM_PARTIAL_RENDER		Section 5.5.1.66
65Ch	EUR_CR_DPM_LSS_PARTIAL_CONTEXT		Section 5.5.1.67
660h	EUR_CR_DPM_START_OF_3D_CONTEXT_PAGES_ALLOCATED		Section 5.5.1.68
680h	EUR_CR_DPM_TASK_3D_FREE		Section 5.5.1.69
684h	EUR_CR_DPM_TASK_TA_FREE		Section 5.5.1.70
694h	EUR_CR_DPM_TASK_STATE	This register controls the DPM module task state for page tables. The normal order of operation is Store context, Clear context, Load new context.	Section 5.5.1.71
698h	EUR_CR_DPM_TASK_CONTROL	This register controls the DPM module task state for control page tables. The normal order of operation is Store context, Clear context, Load new context.	Section 5.5.1.72
69Ch	EUR_CR_DPM_OUTOFMEM		Section 5.5.1.73
6A0h	EUR_CR_DPM_FREE_CONTEXT		Section 5.5.1.74
6A4h	EUR_CR_DPM_3D_TIMEOUT		Section 5.5.1.75
6A8h	EUR_CR_DPM_TA_EVM		Section 5.5.1.76
700h	EUR_CR_DPM_TA_ALLOC_FREE_LIST_STATUS1	This status register is read for a context switch, and reprogrammed on a context load.	Section 5.5.1.77
70Ch	EUR_CR_DPM_3D_FREE_LIST_STATUS1	This status register is read for a context switch, and reprogrammed on a context load.	Section 5.5.1.78
710h	EUR_CR_DPM_TA_ALLOC_FREE_LIST_STATUS2	This status register is read for a context switch, and reprogrammed on a context load.	Section 5.5.1.79
71Ch	EUR_CR_DPM_3D_FREE_LIST_STATUS2	This status register is read for a context switch, and reprogrammed on a context load.	Section 5.5.1.80
720h	EUR_CR_DPM_ABORT_STATUS_MTILE	This status register is read for a context switch, and reprogrammed on a context load.	Section 5.5.1.81
724h	EUR_CR_DPM_PAGE_STATUS	This status register is read for a context switch, and reprogrammed on a context load.	Section 5.5.1.82
728h	EUR_CR_DPM_PAGE	This status register is read for a context switch, and reprogrammed on a context load.	Section 5.5.1.83
72Ch	EUR_CR_DPM_GLOBAL_PAGE_STATUS	This status register is read for a context switch, and reprogrammed on a context load.	Section 5.5.1.84
730h	EUR_CR_DPM_GLOBAL_PAGE	This status register is read for a context switch, and reprogrammed on a context load.	Section 5.5.1.85
734h	EUR_CR_DPM_REQUESTING		Section 5.5.1.86
738h	EUR_CR_DPM_RNG_HDR_PARSER		Section 5.5.1.87
14h	EUR_CR_CORE_REVISION	Core Revision Register identifies the specific core revision. This is updated to reflect the formal release status of the core.	Section 5.5.1.88
800h	EUR_CR_TSP_CONFIG		Section 5.5.1.89
804h	EUR_CR_CACHE_CTRL	Unified Cache global control register	Section 5.5.1.90
18h	EUR_CR_DESIGNER_REV_FIELD1	Designer Revision Field. The SOC designer can use this register for their own revision control if required.	Section 5.5.1.91

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 5-1. SGX530_MMAV10GS70 REGISTERS (continued)

Offset	Acronym	Register Description	Section
100h	EUR_CR_TRIGGER	TA/3D Lockup counter timeout value (16K cycle granularity) (Debug only, for further information contact Imagination Technologies)	Section 5.5.1.92
A00h	EUR_CR_USE_CTRL		Section 5.5.1.93
A04h	EUR_CR_USE_LD_REDIRECT		Section 5.5.1.94
A08h	EUR_CR_USE_CACHE		Section 5.5.1.95
A0Ch	EUR_CR_USE_CODE_BASE00		Section 5.5.1.96
A10h	EUR_CR_USE_CODE_BASE01		Section 5.5.1.97
A14h	EUR_CR_USE_CODE_BASE02		Section 5.5.1.98
A18h	EUR_CR_USE_CODE_BASE03		Section 5.5.1.99
A1Ch	EUR_CR_USE_CODE_BASE04		Section 5.5.1.100
A20h	EUR_CR_USE_CODE_BASE05		Section 5.5.1.101
A24h	EUR_CR_USE_CODE_BASE06		Section 5.5.1.102
A28h	EUR_CR_USE_CODE_BASE07		Section 5.5.1.103
A2Ch	EUR_CR_USE_CODE_BASE08		Section 5.5.1.104
A30h	EUR_CR_USE_CODE_BASE09		Section 5.5.1.105
A34h	EUR_CR_USE_CODE_BASE10		Section 5.5.1.106
A38h	EUR_CR_USE_CODE_BASE11		Section 5.5.1.107
A3Ch	EUR_CR_USE_CODE_BASE12		Section 5.5.1.108
A40h	EUR_CR_USE_CODE_BASE13		Section 5.5.1.109
A44h	EUR_CR_USE_CODE_BASE14		Section 5.5.1.110
A48h	EUR_CR_USE_CODE_BASE15		Section 5.5.1.111
A50h	EUR_CR_EVENT_PDS_ENABLE2	Enables general event bits into the PDS event data master for processing by the PDS Micro Controller	Section 5.5.1.112
A54h	EUR_CR_PDS_CACHE_ENABLE	Enables PDS cache invalidation events bits into the PDS event data master for processing by the PDS Micro Controller	Section 5.5.1.113
A58h	EUR_CR_EVENT_PDS_ENABLE	Enables general event bits into the PDS event data master for processing by the PDS Micro Controller	Section 5.5.1.114
A5Ch	EUR_CR_EVENT_PIXEL_PDS_EXEC		Section 5.5.1.115
A60h	EUR_CR_EVENT_PIXEL_PDS_DATA		Section 5.5.1.116
A64h	EUR_CR_EVENT_PIXEL_PDS_INFO		Section 5.5.1.117
A68h	EUR_CR_EVENT_OTHER_PDS_EXEC		Section 5.5.1.118
A6Ch	EUR_CR_EVENT_OTHER_PDS_DATA		Section 5.5.1.119
A70h	EUR_CR_EVENT_OTHER_PDS_INFO		Section 5.5.1.120
A74h	EUR_CR_DMS_CTRL	Control register used to disable individual data masters - '1' - disable data master, '0' - enable data master	Section 5.5.1.121
A78h	EUR_CR_DMS_AGE		Section 5.5.1.122
A7Ch	EUR_CR_USE_G0		Section 5.5.1.123
A80h	EUR_CR_USE_G1		Section 5.5.1.124
A84h	EUR_CR_USE_FILTER0_LEFT		Section 5.5.1.125
A88h	EUR_CR_USE_FILTER0_RIGHT		Section 5.5.1.126
A8Ch	EUR_CR_USE_FILTER0_EXTRA		Section 5.5.1.127
A90h	EUR_CR_USE_FILTER1_LEFT		Section 5.5.1.128
A94h	EUR_CR_USE_FILTER1_RIGHT		Section 5.5.1.129

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 5-1. SGX530_MMAV10GS70 REGISTERS (continued)

Offset	Acronym	Register Description	Section
A98h	EUR_CR_USE_FILTER1_EXTRA		Section 5.5.1.130
A9Ch	EUR_CR_USE_FILTER_TABLE		Section 5.5.1.131
110h	EUR_CR_EVENT_HOST_ENABLE 2	This register enables interrupts. Writing a '1' to a bit field enables the relevant Event. All the bit fields correspond exactly to those in the EUR_CR_EVENT_STATUS2 register. The MASTER_INTERRUPT bit is a global enable which overrides the Event enables, i.e. '1' - enable external interrupts, '0' - disable external interrupts.	Section 5.5.1.132
AA0h	EUR_CR_USE_ST_RANGE		Section 5.5.1.133
AA4h	EUR_CR_USE0_DM_SLOT		Section 5.5.1.134
AA8h	EUR_CR_USE1_DM_SLOT		Section 5.5.1.135
AB4h	EUR_CR_USE_EXEF		Section 5.5.1.136
AB8h	EUR_CR_PDS_EXEC_BASE	Base Address in memory where the PDS programs are located	Section 5.5.1.137
ABCh	EUR_CR_PDS		Section 5.5.1.138
AC4h	EUR_CR_EVENT_KICKER	Base Address in memory where the Kick Synchronisation data will be read from	Section 5.5.1.139
114h	EUR_CR_EVENT_HOST_CLEAR2	This register is used to clear event interrupts. Writing a '1' to a bit field clears the relevant Event. All the bit fields correspond exactly to those in the EUR_CR_EVENT_STATUS2 register.	Section 5.5.1.140
AC8h	EUR_CR_EVENT_KICK		Section 5.5.1.141
ACCh	EUR_CR_EVENT_TIMER	The timer which runs at the core clock when enabled, decrementing from the loaded value. Wraps when it hits 0 back to the starting value. A transition through 0 creates an event, which can be enabled into either the PDS or the Host status register. The timer can also be read/written from within the PDS program.	Section 5.5.1.142
AD0h	EUR_CR_PDS_INV0		Section 5.5.1.143
AD4h	EUR_CR_PDS_INV1		Section 5.5.1.144
AD8h	EUR_CR_PDS_INV2		Section 5.5.1.145
ADCh	EUR_CR_PDS_INV3		Section 5.5.1.146
AE0h	EUR_CR_PDS_INV_CSC		Section 5.5.1.147
AE4h	EUR_CR_USE_GLOBCOM0		Section 5.5.1.148
AE8h	EUR_CR_USE_GLOBCOM1		Section 5.5.1.149
AECh	EUR_CR_USE_GLOBCOM2		Section 5.5.1.150
1Ch	EUR_CR_DESIGNER_REV_FIEL D2	Designer Revision Field The SOC designer can use this register for their own revision control if required.	Section 5.5.1.151
118h	EUR_CR_EVENT_STATUS2	There are 2 sources of external interrupts in the PowerVR SGX core: General system events and PDS cache invalidation events. The event status2 register indicates the source of any general event interrupt generated by PowerVR SGX. These events only result in an external interrupt if the relevant bit in the EUR_CR_EVENT_HOST_ENABLE register is set. Note1: This register must be combined with the EUR_CR_PDS_CACHE_STATUS register information to determine what has caused the external interrupt. Note2: The host can write to this register in which case any bits written as a '1' are ORed into the register i.e. the host can cause the core to generate an interrupt. (This is typically for debug purposes only)	Section 5.5.1.152
AF0h	EUR_CR_USE_GLOBCOM3		Section 5.5.1.153
AF4h	EUR_CR_USE_GLOBCOM4		Section 5.5.1.154
AF8h	EUR_CR_USE_GLOBCOM5		Section 5.5.1.155
AFCh	EUR_CR_USE_GLOBCOM6		Section 5.5.1.156

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 5-1. SGX530_MMAV10GS70 REGISTERS (continued)

Offset	Acronym	Register Description	Section
B00h	EUR_CR_USE_GLOBCOM7		Section 5.5.1.157
B04h	EUR_CR_LOOPBACK	This USE can issue loopback tasks through the PDS back to itself. This bit indicates the status of this interface. A value of 1 should that there are no pending loopback requests (for any USE pipeline), A value of 0 indicates that a loop task is pending, and that further loopback events may cause the USE pipeline to wait until new resource becomes available. Issuing loopback events when tasks are already pending is not recommended, as this may result in system lockups.	Section 5.5.1.158
B08h	EUR_CR_USE0_SERV_PIXEL		Section 5.5.1.159
B0Ch	EUR_CR_USE0_SERV_VERTEX		Section 5.5.1.160
B10h	EUR_CR_USE0_SERV_EVENT		Section 5.5.1.161
B14h	EUR_CR_USE1_SERV_PIXEL		Section 5.5.1.162
B18h	EUR_CR_USE1_SERV_VERTEX		Section 5.5.1.163
B1Ch	EUR_CR_USE1_SERV_EVENT		Section 5.5.1.164
B20h	EUR_CR_USE_FILTER2_LEFT		Section 5.5.1.165
B24h	EUR_CR_USE_FILTER2_RIGHT		Section 5.5.1.166
B28h	EUR_CR_USE_FILTER2_EXTRA		Section 5.5.1.167
B2Ch	EUR_CR_PDS_PC_BASE		Section 5.5.1.168
B30h	EUR_CR_USE_MOE		Section 5.5.1.169
12Ch	EUR_CR_EVENT_STATUS	There are 2 sources of external interrupts in the PowerVR SGX core: General system events and PDS cache invalidation events. The event status register indicates the source of any general event interrupt generated by PowerVR SGX. These events only result in an external interrupt if the relevant bit in the EUR_CR_EVENT_HOST_ENABLE register is set. Note1: This register must be combined with the EUR_CR_PDS_CACHE_STATUS register information to determine what has caused the external interrupt. Note2: The host can write to this register in which case any bits written as a '1' are ORed into the register i.e. the host can cause the core to generate an interrupt. (This is typically for debug purposes only)	Section 5.5.1.170
130h	EUR_CR_EVENT_HOST_ENABLE	This register enables interrupts. Writing a '1' to a bit field enables the relevant Event. All the bit fields correspond exactly to those in the EUR_CR_EVENT_STATUS register. The MASTER_INTERRUPT bit is a global enable which overrides the Event enables, i.e. '1' - enable external interrupts, '0' - disable external interrupts.	Section 5.5.1.171
C00h	EUR_CR_BIF_CTRL		Section 5.5.1.172
C04h	EUR_CR_BIF_INT_STAT		Section 5.5.1.173
134h	EUR_CR_EVENT_HOST_CLEAR	This register is used to clear event interrupts. Writing a '1' to a bit field clears the relevant Event. All the bit fields correspond exactly to those in the EUR_CR_EVENT_STATUS register.	Section 5.5.1.174
C08h	EUR_CR_BIF_FAULT		Section 5.5.1.175

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 5-1. SGX530_MMAV10GS70 REGISTERS (continued)

Offset	Acronym	Register Description	Section
138h	EUR_CR_PDS_CACHE_STATUS	There are 2 sources of external interrupts in the PowerVR SGX core: General system events and PDS cache invalidation events. The PDS cache status register indicates the source of any PDS cache invalidation event interrupts generated by PowerVR SGX. These events only result in an external interrupt if the relevant bit in the EUR_CR_PDS_CACHE_HOST_ENABLE register is set. Note1: This register must be combined with the EUR_CR_EVENT_STATUS register information to determine what has caused the external interrupt. Note2: The host can write to this register in which case any bits written as a '1' are ORed into the register i.e. the host can cause the core to generate an interrupt. (This is typically for debug purposes only)	Section 5.5.1.176
13Ch	EUR_CR_PDS_CACHE_HOST_ENABLE	This register enables interrupts. Writing a '1' to a bit field enables the relevant Event. All the bit fields correspond exactly to those in the EUR_CR_PDS_CACHE_STATUS register. The MASTER_INTERRUPT bit contained in the EUR_CR_EVENT_HOST_ENABLE register is a global enable which overrides the Event enables, i.e. '1' - enable external interrupts, '0' - disable external interrupts.	Section 5.5.1.177
140h	EUR_CR_PDS_CACHE_HOST_CLEAR	This register is used to clear event interrupts. Writing a '1' to a bit field clears the relevant Event. All the bit fields correspond exactly to those in the EUR_CR_PDS_CACHE_STATUS register.	Section 5.5.1.178
C84h	EUR_CR_BIF_DIR_LIST_BASE0		Section 5.5.1.179
C90h	EUR_CR_BIF_TA_REQ_BASE		Section 5.5.1.180
CA8h	EUR_CR_BIF_MEM_REQ_STAT		Section 5.5.1.181
CACh	EUR_CR_BIF_3D_REQ_BASE		Section 5.5.1.182
CB0h	EUR_CR_BIF_ZLS_REQ_BASE		Section 5.5.1.183
4h	EUR_CR_CLKGATESTATUS	Clock Gating Status reflects the condition of the clock gate controls for each module.	Section 5.5.1.184
200h	EUR_CR_VDM_START	Any write to this register starts the Vertex Data Master DMA operation	Section 5.5.1.185
204h	EUR_CR_TE_AA	This register controls the anti-aliasing mode of the Tiling Co-Processor, independent control is provided in both X and Y axis.	Section 5.5.1.186
208h	EUR_CR_TE_MTILE1	The Tiling Co-Processor allocates memory in macrotiles, which is a group of individual tiles. The number of macrotiles is selectable between 4 and 16, the boundaries of the macrotiles in the X axis are defined in this register.	Section 5.5.1.187
20Ch	EUR_CR_TE_MTILE2	The Tiling Co-Processor allocates memory in macrotiles, which is a group of individual tiles. The number of macrotiles is selectable between 4 and 16, the boundaries of the macrotiles in the Y axis are defined in this register.	Section 5.5.1.188
210h	EUR_CR_TE_SCREEN	In order to perform the tiling operation and generate the display list the maximum screen size must be configured in terms of the number of tiles in X and Y axis.	Section 5.5.1.189
214h	EUR_CR_TE_MTILE	This register defines the number of individual tiles within the macrotiles. This is used in the process of memory allocation.	Section 5.5.1.190
218h	EUR_CR_TE_PSG	This register defines the global control for the Parameter Stream Generator within the Tiling Co-Processor. This module formats the display list generated by the Tiling Co-Processor.	Section 5.5.1.191

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 5-1. SGX530_MMAV10GS70 REGISTERS (continued)

Offset	Acronym	Register Description	Section
21Ch	EUR_CR_TE_PSGREGION_BASE	This register defines the base address in memory of the Region Header writes by the TA. Region headers are the first part of the display list and contain an entry per tile with information on global setup and a link address to parameters.	Section 5.5.1.192
220h	EUR_CR_TE_TPC_BASE	This register defines the base address in memory of the Tail Pointer Cache. A tail pointer is the current last address written to for a tiles individual display list, an entry for active tiles is maintained as primitives are processed by the TA.	Section 5.5.1.193
224h	EUR_CR_TE_TPCCONTROL	The Tail Pointer Cache is used to keep track of the last address written to for a particular tile, this is stored in a mixture of on chip cache and external memory. Under certain circumstances it is necessary to either reset the cache or flush the on chip cache contents such that they are visible in memory.	Section 5.5.1.194
228h	EUR_CR_TE_RGNBBOX_X	Reset or enabled by the MTE_RESETBBOX and MTE_UPDATEBBOX fields in the MTEControl word of the Input Parameter format. This is the maximum extent in X of the post transformed, clipped data before region clipping.	Section 5.5.1.195
22Ch	EUR_CR_TE_RGNBBOX_Y	Reset or enabled by the MTE_RESETBBOX and MTE_UPDATEBBOX fields in the MTE Control word of the Input Parameter format. This is the maximum extent in Y of the post transformed, clipped data before region clipping.	Section 5.5.1.196
230h	EUR_CR_MTE_OTPM_CSM_FLUSH_BASE	The TA maintains internal state of the current information for each macro tile and global list regarding the vertex blocks written to memory. In certain circumstances during a context switch it is necessary to flush this information to external memory.	Section 5.5.1.197
234h	EUR_CR_MTE_OTPM_CSM_LOAD_BASE	The TA maintains internal state of the current information for each macro tile and global list regarding the vertex blocks written to memory. In certain circumstances during a context switch it is necessary to load this information from external memory.	Section 5.5.1.198
238h	EUR_CR_VDM_CTRL_STREAM_BASE	PowerVR SGX masters information from memory using advanced DMA, this register defines the base address of the Vertex Data input control stream in memory.	Section 5.5.1.199
23Ch	EUR_CR_MTE_CTRL	This register controls the global setup of the Tiling Process.	Section 5.5.1.200
240h	EUR_CR_MTE_WCOMPARE		Section 5.5.1.201
244h	EUR_CR_MTE_WCLAMP		Section 5.5.1.202
248h	EUR_CR_MTE_SCREEN	In order to perform the tiling operation and generate the display list the maximum screen size must be configured in terms of the number of pixels in X and Y axis since this may not be the same as the number of tiles defined in the EUR_CR_TE_SCREEN register.	Section 5.5.1.203
24Ch	EUR_CR_MTE_OTPM_OP	The TA maintains internal state of the current information for each macro tile and global list regarding the vertex blocks written to memory. In certain circumstances during a context switch it is necessary to load or flush this information to external memory.	Section 5.5.1.204
250h	EUR_CR_MTE_MULTISAMPLECTL	This applies when anti-aliasing is enabled. This is used to calculate whether an anti-aliased triangle can be perfect small object culled.	Section 5.5.1.205
26Ch	EUR_CR_MTE_SIG1		Section 5.5.1.206
270h	EUR_CR_MTE_SIG2		Section 5.5.1.207
274h	EUR_CR_TE_STATE	This register is read on VDM_START_PULSE and should be saved and updated by the driver app by app	Section 5.5.1.208

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 5-1. SGX530_MMAV10GS70 REGISTERS (continued)

Offset	Acronym	Register Description	Section
40h	EUR_CR_PERF	Within SGX there are a number of performance counters which enable profiling of an application in terms of the low level hardware performance; this register enables selection of specific groups of performance counters and the ability to reset those counters. Full details can be found in the performance profiling section of this document. Write a '1' followed by a '0' to the _CLR fields to reset the specific counter	Section 5.5.1.209
44h	EUR_CR_PERF_COUNTER0	Performance counter 0 status register	Section 5.5.1.210
48h	EUR_CR_PERF_COUNTER1	Performance counter 1 status register	Section 5.5.1.211
4Ch	EUR_CR_PERF_COUNTER2	Performance counter 2 status register	Section 5.5.1.212
304h	EUR_CR_CLIP_SIG1		Section 5.5.1.213
308h	EUR_CR_PIXELBE_SIG01		Section 5.5.1.214
30Ch	EUR_CR_PIXELBE_SIG02		Section 5.5.1.215
8h	EUR_CR_CLKGATECTLOVR	Core module clock gating override controls: allows clocks to be enabled temporarily for register writes by hosts.	Section 5.5.1.216
50h	EUR_CR_PERF_COUNTER3	Performance counter 3 status register	Section 5.5.1.217
330h	EUR_CR_VDM_CONTEXT_STORE_START	Start VDM Context store, pulse kicks off the store operation. Status flags when this is complete and the stream and index position registers are valid	Section 5.5.1.218
334h	EUR_CR_VDM_CONTEXT_STORE_STREAM	Stores the stream position when the context store occured	Section 5.5.1.219
338h	EUR_CR_VDM_CONTEXT_STORE_INDEX	Stores the index position when the context store occured	Section 5.5.1.220
33Ch	EUR_CR_VDM_CONTEXT_STORE_STATUS	Signals when a context store is complete and the stream and index location registers are valid	Section 5.5.1.221
340h	EUR_CR_VDM_CONTEXT_STORE_STATE0	Terminate state control registers used for creating the terminate PDS task.	Section 5.5.1.222
344h	EUR_CR_VDM_CONTEXT_STORE_STATE1	Terminate state control registers used for creating the terminate PDS task.	Section 5.5.1.223
54h	EUR_CR_PERF_COUNTER4	Performance counter 4 status register	Section 5.5.1.224
348h	EUR_CR_VDM_WAIT_FOR_KICK	Indicates the status on the control stream fetch	Section 5.5.1.225
34Ch	EUR_CR_VDM_CONTEXT_STORE_WRAP_INDEX	Stores the number of indices since the last wrap	Section 5.5.1.226
350h	EUR_CR_VDM_CONTEXT_STORE_WRAPPED	Indicates the number of times that index wrapping has occurred after a context store	Section 5.5.1.227
354h	EUR_CR_VDM_CONTEXT_STORE_FAN	The driver sets this register on a context switch such that the hardware can resume knowing where the centre point of a fan is located or the winding order of the strip, expressed as an index.	Section 5.5.1.228
358h	EUR_CR_VDM_CONTEXT_STORE_WRAP	Indicates the number of times that index wrapping has occurred after a context store	Section 5.5.1.229
58h	EUR_CR_PERF_COUNTER5	Performance counter 5 status register	Section 5.5.1.230
5Ch	EUR_CR_PERF_COUNTER6	Performance counter 6 status register	Section 5.5.1.231
60h	EUR_CR_PERF_COUNTER7	Performance counter 7 status register	Section 5.5.1.232

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.1 EUR_CR_CLKGATECTL Register (offset = 0h)

Core module clock gating controls: allows clocks to be forced off, forced on or operate under automatic pipeline activity based clock gating.

EUR_CR_CLKGATECTL is shown in Figure 5-1 and described in Table 5-2.

Figure 5-1. EUR_CR_CLKGATECTL Register

31	30	29	28	27	26	25	24
Reserved							AUTO_MAN_REG
R/W-00h							R/W-0h

23	22	21	20	19	18	17	16
Reserved		USE_CLKG		Reserved		DPM_CLKG	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

15	14	13	12	11	10	9	8
Reserved		TA_CLKG		Reserved		TSP_CLKG	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

7	6	5	4	3	2	1	0
Reserved		ISP_CLKG		Reserved		Reserved	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-2. EUR_CR_CLKGATECTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	Reserved	R/W	00h	
24	AUTO_MAN_REG	R/W	0h	Automatic Register Power Management Control 0 - Register: Register updates automatically turn on clock domains 1 - Manual: Manual clock domain control for register updates
23-22	Reserved	R/W	0h	
21-20	USE_CLKG	R/W	0h	Clock gating control for the USSE 0 - Gated: Clock gating is active and no clock is supplied to the USSE 1 - Running: Clock to the USSE is enabled 2 - Automatic: Automatic gating is active, the USSE clock is only active whilst data is being processed 3 - Reserved_3: Reserved
19-18	Reserved	R/W	0h	
17-16	DPM_CLKG	R/W	0h	Clock gating control for the DPM 0 - Gated: Clock gating is active and no clock is supplied to the DPM 1 - Running: Clock to the DPM is enabled 2 - Automatic: Automatic gating is active, the DPM clock is only active whilst data is being processed 3 - Reserved_3: Reserved
15-14	Reserved	R/W	0h	
13-12	TA_CLKG	R/W	0h	Clock gating control for the TA 0 - Gated: Clock gating is active and no clock is supplied to the TA 1 - Running: Clock to the TA is enabled 2 - Automatic: Automatic gating is active, the TA clock is only active whilst data is being processed 3 - Reserved_3: Reserved
11-10	Reserved	R/W	0h	
9-8	TSP_CLKG	R/W	0h	Clock gating control for the TSP 0 - Gated: Clock gating is active and no clock is supplied to the TSP 1 - Running: Clock to the TSP is enabled 2 - Automatic: Automatic gating is active, the TSP clock is only active whilst data is being processed 3 - Reserved_3: Reserved
7-6	Reserved	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 5-2. EUR_CR_CLKGATECTL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
5-4	ISP_CLKG	R/W	0h	Clock gating control for the ISP 0 - Gated: Clock gating is active and no clock is supplied to the ISP 1 - Running: Clock to the ISP is enabled 2 - Automatic: Automatic gating is active, the ISP clock is only active whilst data is being processed 3 - Reserved_3: Reserved
3-2	Reserved	R/W	0h	
1-0	Reserved	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.2 EUR_CR_PERF_COUNTER8 Register (offset = 64h)

Performance counter status register

EUR_CR_PERF_COUNTER8 is shown in Figure 5-2 and described in Table 5-3.

Figure 5-2. EUR_CR_PERF_COUNTER8 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-3. EUR_CR_PERF_COUNTER8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COUNTER8	R	00000000h	32 bit integer count value, see performance profiling section for details

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.3 EUR_CR_PIXELBE Register (offset = 400h)

The PixelBE modules format the pixel data before writing to memory, this register contains global control signals.

EUR_CR_PIXELBE is shown in Figure 5-3 and described in Table 5-4.

Figure 5-3. EUR_CR_PIXELBE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-4. EUR_CR_PIXELBE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R/W	000000h	
8	IDF_USE_FBADDR	R/W	0h	Use Frame buffer address for IDF's when set to 1. Default is 0 which means IDFs use the last address written too by the PBE
7-0	ALPHATHRESHOLD	R/W	00h	Alpha threshold used when encoding 1555 format. If the internal 8bit Alpha value exceeds this value, then a 1 is written to the top bit of the output, otherwise a 0 is written

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.4 EUR_CR_ISP_RENDER Register (offset = 404h)

The Image Synthesis Processor can be configured to operate in 3 different render modes

EUR_CR_ISP_RENDER is shown in Figure 5-4 and described in Table 5-5.

Figure 5-4. EUR_CR_ISP_RENDER Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-5. EUR_CR_ISP_RENDER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	Reserved	R/W	00000000h	
1-0	TYPE	R/W	0h	Render type 0 - 3D_0: = Normal 3D render 1 - 3D_1: = Normal 3D render 2 - 2D: = Fast 2D render 3 - Scale: = Fast scale render

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.5 EUR_CR_ISP_RGN_BASE Register (offset = 408h)

This register defines the base address in memory of the Region Header reads by the ISP. Region headers are the first part of the display list and contain an entry per tile with information on global setup and a link address to parameters.

EUR_CR_ISP_RGN_BASE is shown in Figure 5-5 and described in Table 5-6.

Figure 5-5. EUR_CR_ISP_RGN_BASE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-6. EUR_CR_ISP_RGN_BASE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved	R/W	0h	
27-2	ADDR	R/W	0000000h	256MB Addressable, Dword aligned Base Address for Region Header Reads
1-0	Reserved	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.6 EUR_CR_ISP_RENDBOX1 Register (offset = 40Ch)

When the render type is set to Fast 2D Render or Fast Scale Render, no region headers are read from memory. Instead, region headers are internally generated for all tiles enclosed in the bounding box defined by this register and the EUR_CR_ISP_RENDBOX2 register.

EUR_CR_ISP_RENDBOX1 is shown in Figure 5-6 and described in Table 5-7.

Figure 5-6. EUR_CR_ISP_RENDBOX1 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-7. EUR_CR_ISP_RENDBOX1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	Reserved	R/W	00h	
23-16	X	R/W	00h	Bounding box X1. X address, in tiles, of top left most tile in bounding box. The LSB is ignored if AA_MODE is low.
15-8	Reserved	R/W	00h	
7-0	Y	R/W	00h	Bounding box Y1. Y address, in tiles, of top left most tile in bounding box. The LSB is ignored if AA_MODE is low.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.7 EUR_CR_ISP_RENDBOX2 Register (offset = 410h)

When the render type is set to Fast 2D Render or Fast Scale Render, no region headers are read from memory. Instead, region headers are internally generated for all tiles enclosed in the bounding box defined by this register and the EUR_CR_ISP_RENDBOX1 register.

EUR_CR_ISP_RENDBOX2 is shown in Figure 5-7 and described in Table 5-8.

Figure 5-7. EUR_CR_ISP_RENDBOX2 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-8. EUR_CR_ISP_RENDBOX2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	Reserved	R/W	00h	
23-16	X	R/W	00h	Bounding box X2. X address, in tiles, of bottom right most tile in bounding box. The LSB is ignored if AA_MODE is low.
15-8	Reserved	R/W	00h	
7-0	Y	R/W	00h	Bounding box Y2. Y address, in tiles, of bottom right most tile in bounding box.The LSB is ignored if AA_MODE is low.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.8 EUR_CR_ISP_IPFMISC Register (offset = 414h)

Global control register for the ISP parameter fetch.

EUR_CR_ISP_IPFMISC is shown in Figure 5-8 and described in Table 5-9.

Figure 5-8. EUR_CR_ISP_IPFMISC Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-9. EUR_CR_ISP_IPFMISC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	R/W	00h	
25-24	TTE_SCALE	R/W	0h	Scale factor for ISP parameter fetch to BIF TTE.
23-20	Reserved	R/W	0h	
19-16	UPASSSTART	R/W	0h	Initial upass_ctrl value, loaded at the start of each tile, for user pass spawning in the ISP.
15-9	Reserved	R/W	00h	
8	PROCESSEMPTY	R/W	0h	If this bit is set, all empty regions encountered will be processed by the 3D core ie the hardware background object being rendered in each empty tile. NOTE that this bit does not affect a "Fast Scale Render" i.e. empty regions are always processed when performing a "Fast Scale Render".
7-6	Reserved	R/W	0h	
5-0	VALIDID	R/W	00h	6 bit field specify the minimum object ID required for opaque and translucent objects to be rendered.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.9 EUR_CR_ISP_OGL_MODE Register (offset = 418h)

Global control register which modifies the pixel sample position to comply with OGL requirements.

EUR_CR_ISP_OGL_MODE is shown in Figure 5-9 and described in Table 5-10.

Figure 5-9. EUR_CR_ISP_OGL_MODE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-10. EUR_CR_ISP_OGL_MODE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R/W	00000000h	
0	ENABLE	R/W	0h	0 Non OGL Rules 0 - Disabled: Non OGL Rules 1 - Enabled: OGL Rules

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.10 EUR_CR_ISP_PERPENDICULAR Register (offset = 41Ch)

EUR_CR_ISP_PERPENDICULAR is shown in Figure 5-10 and described in Table 5-11.

Figure 5-10. EUR_CR_ISP_PERPENDICULAR Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-11. EUR_CR_ISP_PERPENDICULAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	Reserved	R/W	0h	
30-4	THRESHOLD	R/W	0000000h	If the determinant in the ISP FPU is smaller than this unsigned float, then perpendicular maths is used.
3-0	Reserved	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.11 EUR_CR_ISP_CULLVALUE Register (offset = 420h)

EUR_CR_ISP_CULLVALUE is shown in Figure 5-11 and described in Table 5-12.

Figure 5-11. EUR_CR_ISP_CULLVALUE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-12. EUR_CR_ISP_CULLVALUE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	Reserved	R/W	0h	
30-4	THRESHOLD	R/W	0000000h	If the determinant in the ISP FPU is smaller than this unsigned float, then the triangle is culled.
3-0	Reserved	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.12 EUR_CR_ISP_DBIAS Register (offset = 424h)

EUR_CR_ISP_DBIAS is shown in Figure 5-12 and described in Table 5-13.

Figure 5-12. EUR_CR_ISP_DBIAS Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-13. EUR_CR_ISP_DBIAS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	Reserved	R/W	0000h	
15-8	FACTORADJ	R/W	00h	Signed 2's complement integer used to control the slope dependent part of depth bias. The default value should be -23.
7-0	UNITSADJ	R/W	00h	Signed 2's complement integer used to control the slope independent part of depth bias. The default value should be 0.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.13 EUR_CR_ISP_START_RENDER Register (offset = 428h)

Any write to this register starts a 3D Render.

EUR_CR_ISP_START_RENDER is shown in Figure 5-13 and described in Table 5-14.

Figure 5-13. EUR_CR_ISP_START_RENDER Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-14. EUR_CR_ISP_START_RENDER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	W	00000000h	
0	PULSE	W	0h	Start 3D render

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.14 EUR_CR_3D_AA_MODE Register (offset = 42Ch)

Indicates to the 3D pipeline whether anti-aliasing is enabled or disabled

EUR_CR_3D_AA_MODE is shown in Figure 5-14 and described in Table 5-15.

Figure 5-14. EUR_CR_3D_AA_MODE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-15. EUR_CR_3D_AA_MODE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R/W	00000000h	
0	ENABLE	R/W	0h	0 - Disabled: anti aliasing disabled 1 - Enabled: anti aliasing enabled

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.15 EUR_CR_ISP_BREAK Register (offset = 430h)

Object breakpoints can be inserted by the driver to interrupt a render part way and then resume the render from the point of interrupt at a later time, this register controls what happens when a breakpoint has been encountered.

EUR_CR_ISP_BREAK is shown in Figure 5-15 and described in Table 5-16.

Figure 5-15. EUR_CR_ISP_BREAK Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-16. EUR_CR_ISP_BREAK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	Reserved	W	0000000h	
4	RESUME	W	0h	Resumes the current render after a breakpoint object is encountered in the internal parameter stream
3-1	Reserved	W	0h	
0	HALT	W	0h	Terminates the current render after a breakpoint object is encountered in the internal parameter stream

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.16 EUR_CR_ISP_3DCONTEXT Register (offset = 434h)

EUR_CR_ISP_3DCONTEXT is shown in Figure 5-16 and described in Table 5-17.

Figure 5-16. EUR_CR_ISP_3DCONTEXT Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-17. EUR_CR_ISP_3DCONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	W	00000000h	
0	ENABLE	W	0h	ISP Context Store, any write to this register will cause the ISP to halt processing new primitives, insert an end of render command and wait for the current primitives in the pipeline to complete processing up to the inserted end of render.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.17 EUR_CR_ISP_FPU Register (offset = 438h)

EUR_CR_ISP_FPU is shown in Figure 5-17 and described in Table 5-18.

Figure 5-17. EUR_CR_ISP_FPU Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-18. EUR_CR_ISP_FPU Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SIGNATURE	R	00000000h	ISP FPU signature result. A signature is an accumulative LFSR (Linear Feedback Shift Register) of the valid pipeline data using an XOR structure.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.18 EUR_CR_TSP_PARAMETER_CACHE Register (offset = 43Ch)

EUR_CR_TSP_PARAMETER_CACHE is shown in Figure 5-18 and described in Table 5-19.

Figure 5-18. EUR_CR_TSP_PARAMETER_CACHE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-19. EUR_CR_TSP_PARAMETER_CACHE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	W	00000000h	
0	INVALIDATE	W	0h	Any write to this register will invalidate the Content Addressable Memory in the TSP parameter fetch cache

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.19 EUR_CR_TSP_PARAMETER_FETCH Register (offset = 440h)

EUR_CR_TSP_PARAMETER_FETCH is shown in Figure 5-19 and described in Table 5-20.

Figure 5-19. EUR_CR_TSP_PARAMETER_FETCH Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-20. EUR_CR_TSP_PARAMETER_FETCH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	Reserved	R/W	00000000h	
1-0	TTE_SCALE	R/W	0h	Scale factor for TSP parameter fetch to BIF TTE.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.20 EUR_CR_ISP_ZLSCTL Register (offset = 480h)

Image Synthesis Processor Z Load/Store and format global control register

EUR_CR_ISP_ZLSCTL is shown in Figure 5-20 and described in Table 5-21.

Figure 5-20. EUR_CR_ISP_ZLSCTL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-21. EUR_CR_ISP_ZLSCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	Reserved	R/W	000h	
22	STOREMASK	R/W	0h	This bit modifies the behaviour when ISP_ZLSCTL:[ZSTOREFORMAT] = 0x0. 0 - Binary 0 is stored in place of the current rendered mask plane. All stored Z values will appear in memory as positive IEEE floats; 1 - The mask plane is stored per pixel location a rendered to the on-chip tile buffer and occupies the MSB (sign bit) of all IEEE float dwords
21	LOADMASK	R/W	0h	This bit modifies the behaviour when ISP_ZLSCTL:[ZLOADFORMAT] = 0x0. 0 - The entire tile will be loaded with the mask plane value as determined by ISP_BGOBJ:[MASK]; 1 - The mask plane is loaded exactly as it appears in memory with the mask plane bit in the MSB (sign bit) of each loaded IEEE float formatted dword
20	ZONLYRENDER	R/W	0h	When set, only the Z buffer is rendered. Opaque and translucent objects are stencil and depth tested as usual but no pixel spans are emitted to the PDS Pixel presenter. Pixels within punch through and depth feedback objects are emitted on their first pass as a result of passing the up-front depth test by default. However, on their feedback depth test, these pixels will not be output regardless of the depth test result. The intention is to calculate a scene Z buffer without doing any further rendering.
19-18	ZSTOREFORMAT	R/W	0h	Selects the format depth, stencil and viewport is stored in: 0x0 32 bit IEEE SP Float; 0x1 24 bit Integer + 8-bit Stencil; 0x2 Two 16 bit Integers (Packed format); 0x3 32-bit IEEE Float 8-bit Stencil 1-bit Viewport Compressed, addressed using DPM
17-16	ZLOADFORMAT	R/W	0h	Selects the format depth, stencil and viewport is stored in: 0x0 32 bit IEEE SP Float; 0x1 24 bit Integer + 8-bit Stencil; 0x2 Two 16 bit Integers (Packed format); 0x3 32-bit IEEE Float 8-bit Stencil 1-bit Viewport Compressed, addressed using DPM
15-8	ZLSEXTENT	R/W	00h	Display width in tiles (16 pixels) minus one: 0x00 1 tile (16 pixels) 0x01 2 tiles (32 pixels) .. 0xFF 256 tiles (4096 pixels)
7	SSTOREEN	R/W	0h	When set to 1, if the ZSTORE bit in the region header is also set then the stencil buffer is stored to memory after each tile is processed
6	FORCEZSTORE	R/W	0h	If set to 1 the depth/stencil buffer is always stored at the end of each tile irrespective of the region header ZSTORE bit.
5	FORCEZLOAD	R/W	0h	If set to 1 the depth/stencil buffer is always loaded at the start of each tile irrespective of the region header ZLOAD bit.
4	ZSTOREEN	R/W	0h	When set to 1, if the ZSTORE bit in the region header is also set then the depth buffer is stored to memory after each tile is processed
3	STORETILED	R/W	0h	When set to 1 depth and stencil data is written out in "Tiled" order. (Note: Data can not be written as tiled, if the ZSTOREFORMAT = 0x3. In this case STORETILED should be set to 0)
2	ZLOADEN	R/W	0h	When set to 1, if the ZLOAD bit in the region header is also set then the depth buffer is read from memory prior to tile processing
1	SLOADEN	R/W	0h	When set to 1, if the ZLOAD bit in the region header is also set then the stencil buffer is read from memory prior to tile processing
0	LOADTILED	R/W	0h	When set to 1 depth, stencil data is loaded in "Tiled" order (Note: Data can not be read as tiled, if the ZSTOREFORMAT = 0x3. In this case LOADTILED should be set to 0)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.21 EUR_CR_ISP_ZLOAD_BASE Register (offset = 484h)

This register defines the base address in memory of the Z Buffer base address to load into the ISP for non-compressed ZLS formats.

EUR_CR_ISP_ZLOAD_BASE is shown in Figure 5-21 and described in Table 5-22.

Figure 5-21. EUR_CR_ISP_ZLOAD_BASE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-22. EUR_CR_ISP_ZLOAD_BASE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	R/W	00h	
25-4	ADDR	R/W	000000h	64MB Addressable, 16byte aligned Base Address of the Z Buffer Load base address
3-0	Reserved	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.22 EUR_CR_ISP_ZSTORE_BASE Register (offset = 488h)

This register defines the base address in memory of the Z Buffer base address to store data from the ISP for non-compressed ZLS formats.

EUR_CR_ISP_ZSTORE_BASE is shown in Figure 5-22 and described in Table 5-23.

Figure 5-22. EUR_CR_ISP_ZSTORE_BASE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-23. EUR_CR_ISP_ZSTORE_BASE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	R/W	00h	
25-4	ADDR	R/W	000000h	64MB Addressable, 16byte aligned Base Address of the Z Buffer Load base address
3-0	Reserved	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.23 EUR_CR_ISP_STENCIL_LOAD_BASE Register (offset = 48Ch)

This register defines the base address in memory of the Stencil Buffer base address to load into the ISP for Int24 and IEEE ZLS formats. This alternate stencil buffer base address is selectable based on the enable bit.

EUR_CR_ISP_STENCIL_LOAD_BASE is shown in Figure 5-23 and described in Table 5-24.

Figure 5-23. EUR_CR_ISP_STENCIL_LOAD_BASE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-24. EUR_CR_ISP_STENCIL_LOAD_BASE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	ENABLE	R/W	0h	When set to 1, enables fetching of stencil from a separate base address
30-26	Reserved	R/W	00h	
25-4	ADDR	R/W	000000h	64MB Addressable, 16byte aligned Base Address of the Stencil Buffer Load base address
3-0	Reserved	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.24 EUR_CR_ISP_STENCIL_STORE_BASE Register (offset = 490h)

This register defines the base address in memory of the Stencil Buffer base address to store data from the ISP for Int24 and IEEE ZLS formats. This alternate stencil buffer base address is selectable based on the enable bit.

EUR_CR_ISP_STENCIL_STORE_BASE is shown in Figure 5-24 and described in Table 5-25.

Figure 5-24. EUR_CR_ISP_STENCIL_STORE_BASE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-25. EUR_CR_ISP_STENCIL_STORE_BASE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	ENABLE	R/W	0h	When set to 1, enables saving of stencil to a separate base address
30-26	Reserved	R/W	00h	
25-4	ADDR	R/W	000000h	64MB Addressable, 16byte aligned Base Address of the Stencil Buffer Load base address
3-0	Reserved	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.25 EUR_CR_ISP_VISTEST_CTRL Register (offset = 494h)

This register can be used to reset the results of the 8 visibility test results. Write a 1 to the required bit field to clear the result register

EUR_CR_ISP_VISTEST_CTRL is shown in Figure 5-25 and described in Table 5-26.

Figure 5-25. EUR_CR_ISP_VISTEST_CTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-26. EUR_CR_ISP_VISTEST_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	Reserved	W	000000h	
7	CLEAR7	W	0h	Resets Visibility Test register 7
6	CLEAR6	W	0h	Resets Visibility Test register 6
5	CLEAR5	W	0h	Resets Visibility Test register 5
4	CLEAR4	W	0h	Resets Visibility Test register 4
3	CLEAR3	W	0h	Resets Visibility Test register 3
2	CLEAR2	W	0h	Resets Visibility Test register 2
1	CLEAR1	W	0h	Resets Visibility Test register 1
0	CLEAR0	W	0h	Resets Visibility Test register 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.26 EUR_CR_ISP_VISTEST_VISIBLE0 Register (offset = 498h)

EUR_CR_ISP_VISTEST_VISIBLE0 is shown in Figure 5-26 and described in Table 5-27.

Figure 5-26. EUR_CR_ISP_VISTEST_VISIBLE0 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-27. EUR_CR_ISP_VISTEST_VISIBLE0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COUNT	R	00000000h	Visibility Test counter 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.27 EUR_CR_ISP_VISTEST_VISIBLE1 Register (offset = 49Ch)

EUR_CR_ISP_VISTEST_VISIBLE1 is shown in Figure 5-27 and described in Table 5-28.

Figure 5-27. EUR_CR_ISP_VISTEST_VISIBLE1 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-28. EUR_CR_ISP_VISTEST_VISIBLE1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COUNT	R	00000000h	Visibility Test counter 1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.28 EUR_CR_ISP_VISTEST_VISIBLE2 Register (offset = 4A0h)

EUR_CR_ISP_VISTEST_VISIBLE2 is shown in Figure 5-28 and described in Table 5-29.

Figure 5-28. EUR_CR_ISP_VISTEST_VISIBLE2 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-29. EUR_CR_ISP_VISTEST_VISIBLE2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COUNT	R	00000000h	Visibility Test counter 2

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.29 EUR_CR_ISP_VISTEST_VISIBLE3 Register (offset = 4A4h)

EUR_CR_ISP_VISTEST_VISIBLE3 is shown in Figure 5-29 and described in Table 5-30.

Figure 5-29. EUR_CR_ISP_VISTEST_VISIBLE3 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-30. EUR_CR_ISP_VISTEST_VISIBLE3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COUNT	R	00000000h	Visibility Test counter 3

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.30 EUR_CR_ISP_VISTEST_VISIBLE4 Register (offset = 4A8h)

EUR_CR_ISP_VISTEST_VISIBLE4 is shown in Figure 5-30 and described in Table 5-31.

Figure 5-30. EUR_CR_ISP_VISTEST_VISIBLE4 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-31. EUR_CR_ISP_VISTEST_VISIBLE4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COUNT	R	00000000h	Visibility Test counter 4

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.31 EUR_CR_ISP_VISTEST_VISIBLE5 Register (offset = 4ACh)

EUR_CR_ISP_VISTEST_VISIBLE5 is shown in Figure 5-31 and described in Table 5-32.

Figure 5-31. EUR_CR_ISP_VISTEST_VISIBLE5 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-32. EUR_CR_ISP_VISTEST_VISIBLE5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COUNT	R	00000000h	Visibility Test counter 5

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.32 EUR_CR_ISP_VISTEST_VISIBLE6 Register (offset = 4B0h)

EUR_CR_ISP_VISTEST_VISIBLE6 is shown in Figure 5-32 and described in Table 5-33.

Figure 5-32. EUR_CR_ISP_VISTEST_VISIBLE6 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-33. EUR_CR_ISP_VISTEST_VISIBLE6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COUNT	R	00000000h	Visibility Test counter 6

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.33 EUR_CR_ISP_VISTEST_VISIBLE7 Register (offset = 4B4h)

EUR_CR_ISP_VISTEST_VISIBLE7 is shown in Figure 5-33 and described in Table 5-34.

Figure 5-33. EUR_CR_ISP_VISTEST_VISIBLE7 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-34. EUR_CR_ISP_VISTEST_VISIBLE7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COUNT	R	00000000h	Visibility Test counter 7

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.34 EUR_CR_ISP_BGOBJDEPTH Register (offset = 4B8h)

The ISP operates by comparing depth values of incoming objects with the results of previous depth compares, in order to make sure there are no un-initialised values at the start of the tile or to cover pixels where there are no objects in the scene a default background object is configured under register control. This register provides the floating point depth value for this background object.

EUR_CR_ISP_BGOBJDEPTH is shown in Figure 5-34 and described in Table 5-35.

Figure 5-34. EUR_CR_ISP_BGOBJDEPTH Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-35. EUR_CR_ISP_BGOBJDEPTH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VALUE	R/W	00000000h	Hardware background object depth value in 32 bit IEEE single precision floating point format

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.35 EUR_CR_ISP_BGOBJ Register (offset = 4BCh)

The ISP operates by comparing depth values of incoming objects with the results of previous depth compares, in order to make sure there is no un-initialised values at the start of the tile or to cover pixels where there are no objects in the scene a default background object is configured under register control. This register provides tag, mask and stencil information for this background object.

EUR_CR_ISP_BGOBJ is shown in Figure 5-35 and described in Table 5-36.

Figure 5-35. EUR_CR_ISP_BGOBJ Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-36. EUR_CR_ISP_BGOBJ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	Reserved	R/W	000000h	
9	ENABLEBGTAG	R/W	0h	When set to 1, at the start of each tile the ISP tag buffer is initialised with the background object tag (default = 1)
8	MASK	R/W	0h	Hardware background object mask plane
7-0	STENCIL	R/W	00h	Hardware background object stencil

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.36 EUR_CR_ISP_BGOBJTAG Register (offset = 4C4h)

The ISP operates by comparing depth values of incoming objects with the results of previous depth compares, in order to make sure there are no unitialised values at the start of the tile or to cover pixels where there are no objects in the scene a default background object is configured under register control. This register provides TSP and Vertex information for this background object.

EUR_CR_ISP_BGOBJTAG is shown in Figure 5-36 and described in Table 5-37.

Figure 5-36. EUR_CR_ISP_BGOBJTAG Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-37. EUR_CR_ISP_BGOBJTAG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	Reserved	R/W	0h	
29-24	TSPDATASIZE	R/W	00h	Hardware background object TSP data size in dwords
23-22	Reserved	R/W	0h	
21-0	VERTEXPTR	R/W	000000h	Bits 25:4 of the hardware background object vertex pointer (64MB range on 16 byte boundaries)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.37 EUR_CR_ISP_MULTISAMPLECTL Register (offset = 4C8h)

This applies when anti-aliasing is enabled. This is used by the ISP to adjust the sample positions.

EUR_CR_ISP_MULTISAMPLECTL is shown in Figure 5-37 and described in Table 5-38.

Figure 5-37. EUR_CR_ISP_MULTISAMPLECTL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-38. EUR_CR_ISP_MULTISAMPLECTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	MSAA_Y3	R/W	0h	Unsigned sub-pixel offset for the 4th multisample Y position
27-24	MSAA_X3	R/W	0h	Unsigned sub-pixel offset for the 4th multisample X position
23-20	MSAA_Y2	R/W	0h	Unsigned sub-pixel offset for the 3rd multisample Y position
19-16	MSAA_X2	R/W	0h	Unsigned sub-pixel offset for the 3rd multisample X position
15-12	MSAA_Y1	R/W	0h	Unsigned sub-pixel offset for the 2nd multisample Y position
11-8	MSAA_X1	R/W	0h	Unsigned sub-pixel offset for the 2nd multisample X position
7-4	MSAA_Y0	R/W	0h	Unsigned sub-pixel offset for the 1st multisample Y position
3-0	MSAA_X0	R/W	0h	Unsigned sub-pixel offset for the 1st multisample X position

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.38 EUR_CR_ISP_SIG1 Register (offset = 4CCh)

EUR_CR_ISP_SIG1 is shown in Figure 5-38 and described in Table 5-39.

Figure 5-38. EUR_CR_ISP_SIG1 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-39. EUR_CR_ISP_SIG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	PRECALC	R	00000000h	Signature analysis - face equation precalc. A signature is an accumulative LFSR (Linear Feedback Shift Register) of the valid pipeline data using an XOR structure.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.39 EUR_CR_ISP_SIG2 Register (offset = 4D0h)

EUR_CR_ISP_SIG2 is shown in Figure 5-39 and described in Table 5-40.

Figure 5-39. EUR_CR_ISP_SIG2 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-40. EUR_CR_ISP_SIG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	EDGE	R	00000000h	Signature analysis - edge processors. A signature is an accumulative LFSR (Linear Feedback Shift Register) of the valid pipeline data using an XOR structure.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.40 EUR_CR_ISP_SIG3 Register (offset = 4D4h)

EUR_CR_ISP_SIG3 is shown in Figure 5-40 and described in Table 5-41.

Figure 5-40. EUR_CR_ISP_SIG3 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-41. EUR_CR_ISP_SIG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TAGWRITE	R	00000000h	Signature analysis - state buffer storage. A signature is an accumulative LFSR (Linear Feedback Shift Register) of the valid pipeline data using an XOR structure.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.41 EUR_CR_ISP_SIG4 Register (offset = 4D8h)

EUR_CR_ISP_SIG4 is shown in Figure 5-41 and described in Table 5-42.

Figure 5-41. EUR_CR_ISP_SIG4 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-42. EUR_CR_ISP_SIG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SPAN	R	00000000h	Signature analysis - spans to PDS pixel presenter. A signature is an accumulative LFSR (Linear Feedback Shift Register) of the valid pipeline data using an XOR structure.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.42 EUR_CR_ISP_TAGCTRL Register (offset = 4DCh)

EUR_CR_ISP_TAGCTRL is shown in Figure 5-42 and described in Table 5-43.

Figure 5-42. EUR_CR_ISP_TAGCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-43. EUR_CR_ISP_TAGCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	Reserved	R/W	00000000h	
2	SLOWFPT	R/W	0h	Debug punchthrough, converts fast punchthrough to normal punchthrough
1	FORCE_PT_OFF	R/W	0h	Debug punchthrough mode which forces pt_off after every pt,dfb and fpt triangle
0	SAFE	R/W	0h	Safe flush mode, pass-spawn flushes for every translucent object

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.43 EUR_CR_ISP_STATUS1 Register (offset = 4E0h)

EUR_CR_ISP_STATUS1 is shown in Figure 5-43 and described in Table 5-44.

Figure 5-43. EUR_CR_ISP_STATUS1 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-44. EUR_CR_ISP_STATUS1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	R	XXXXh	
25-2	CTRL_STREAM_POS	R	000000h	Control Stream Position after context switch
1-0	Reserved	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.44 EUR_CR_ISP_STATUS2 Register (offset = 4E4h)

EUR_CR_ISP_STATUS2 is shown in Figure 5-44 and described in Table 5-45.

Figure 5-44. EUR_CR_ISP_STATUS2 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-45. EUR_CR_ISP_STATUS2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	Reserved	R	000h	
20-16	PRIM_NUM	R	00h	Position of current primitive after context switch
15-8	TILE_Y	R	00h	Tile Y coordinate after context switch
7-0	TILE_X	R	00h	Tile X coordinate after context switch

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.45 EUR_CR_ISP_DEPTHSORT Register (offset = 4E8h)

EUR_CR_ISP_DEPTHSORT is shown in Figure 5-45 and described in Table 5-46.

Figure 5-45. EUR_CR_ISP_DEPTHSORT Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-46. EUR_CR_ISP_DEPTHSORT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R/W	00000000h	
0	DETERMINISTIC	R/W	0h	deterministic mode for depth sorter

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.46 EUR_CR_PIXELBE_EMIT Register (offset = 4ECh)

This register indicates if the PixelBE would stall the USE if there were another emit. '1' indicates that the PixelBE would stall the USE. '0' indicates that the PixelBE would not stall the USE.

EUR_CR_PIXELBE_EMIT is shown in Figure 5-46 and described in Table 5-47.

Figure 5-46. EUR_CR_PIXELBE_EMIT Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-47. EUR_CR_PIXELBE_EMIT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	Reserved	R	00000000h	
1	STATUS1	R	1h	Pipe 1 status
0	STATUS0	R	1h	Pipe 0 status

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.47 EUR_CR_SOFT_RESET Register (offset = 80h)

Soft reset control register which drives all modules except the register bank. Write a '1' to reset and a '0' to clear

EUR_CR_SOFT_RESET is shown in Figure 5-47 and described in Table 5-48.

Figure 5-47. EUR_CR_SOFT_RESET Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-48. EUR_CR_SOFT_RESET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	Reserved	R/W	0000000h	
6	TSP_RESET	R/W	0h	TSP soft reset control field
5	ISP_RESET	R/W	0h	ISP soft reset control field
4	USE_RESET	R/W	0h	USE soft reset control field
3	TA_RESET	R/W	0h	TA soft reset control field
2	DPM_RESET	R/W	0h	DPM soft reset control field
1	Reserved	R/W	0h	
0	BIF_RESET	R/W	0h	BIF soft reset control field

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.48 EUR_CR_DPM_3D_PAGE_TABLE_BASE Register (offset = 600h)

Effective on load 3D context, this register defines the base address of the page table being referenced in the process of de-allocating pages during a 3D render.

EUR_CR_DPM_3D_PAGE_TABLE_BASE is shown in Figure 5-48 and described in Table 5-49.

Figure 5-48. EUR_CR_DPM_3D_PAGE_TABLE_BASE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-49. EUR_CR_DPM_3D_PAGE_TABLE_BASE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved	R/W	0h	
27-4	ADDR	R/W	000000h	256MB Addressable, 16Byte aligned Base Address for 3D context page tables in the DPM module
3-0	Reserved	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.49 EUR_CR_DPM_3D_FREE_LIST Register (offset = 604h)

Between the FREE_LIST_HEAD and FREE_LIST_TAIL there must be a valid chain of linked pages. The linked list is stored in memory, at the EUR_CR_DPM_3D_PAGE_TABLE_BASE_ADDRESS and is effective on load 3D context.

EUR_CR_DPM_3D_FREE_LIST is shown in Figure 5-49 and described in Table 5-50.

Figure 5-49. EUR_CR_DPM_3D_FREE_LIST Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-50. EUR_CR_DPM_3D_FREE_LIST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	Reserved	R/W	0h	
29-16	TAIL	R/W	0000h	This register defines the last page to be used for the 3D page allocation, on a load 3D context. (The tail of the free list.)
15-14	Reserved	R/W	0h	
13-0	HEAD	R/W	0000h	This register defines the first page to be used for the 3D page allocation, on a load 3D context.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.50 EUR_CR_DPM_PDS_PAGE_THRESHOLD Register (offset = 614h)

This is the threshold in 4KB pages at which the hardware will stop double buffering the internal vertex processing buffers (and drop to single buffered mode). This is required in order to guarantee that a 3D render can run when running in dynamic parameter management mode. It should be set to 0.25MB less (in pages 0x40) than the ZLS threshold (or 0 in very small memory systems).

EUR_CR_DPM_PDS_PAGE_THRESHOLD is shown in Figure 5-50 and described in Table 5-51.

Figure 5-50. EUR_CR_DPM_PDS_PAGE_THRESHOLD Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-51. EUR_CR_DPM_PDS_PAGE_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	Reserved	R/W	00000h	
13-0	VALUE	R/W	2000h	Page Threshold at which vertex buffers stop being double buffered

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.51 EUR_CR_DPM_TA_ALLOC_PAGE_TABLE_BASE Register (offset = 618h)

Effective on Loading the TA Page Table Context, this register defines the base address of the page table being referenced in the process of allocating pages by the TE and MTE.

EUR_CR_DPM_TA_ALLOC_PAGE_TABLE_BASE is shown in Figure 5-51 and described in Table 5-52.

Figure 5-51. EUR_CR_DPM_TA_ALLOC_PAGE_TABLE_BASE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-52. EUR_CR_DPM_TA_ALLOC_PAGE_TABLE_BASE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved	R/W	0h	
27-4	ADDR	R/W	000000h	256MB Addressable, 16Byte aligned Base Address for TE/MTE context page tables in the DPM module.
3-0	Reserved	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.52 EUR_CR_DPM_TA_ALLOC_FREE_LIST Register (offset = 61Ch)

Between the FREE_LIST_HEAD and FREE_LIST_TAIL there must be a valid chain of linked pages. The linked list is stored in memory, at the EUR_CR_DPM_TA_ALLOC_PAGE_TABLE_BASE_ADDRESS and is effective on Loading the TA Page Table Context.

EUR_CR_DPM_TA_ALLOC_FREE_LIST is shown in Figure 5-52 and described in Table 5-53.

Figure 5-52. EUR_CR_DPM_TA_ALLOC_FREE_LIST Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-53. EUR_CR_DPM_TA_ALLOC_FREE_LIST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	Reserved	R/W	0h	
29-16	TAIL	R/W	0000h	This register defines the last page for the free list manipulated during TE and MTE page allocation
15-14	Reserved	R/W	0h	
13-0	HEAD	R/W	0000h	This register defines the start page for the free list manipulated during TE and MTE page allocation

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.53 EUR_CR_DPM_TA_PAGE_THRESHOLD Register (offset = 620h)

EUR_CR_DPM_TA_PAGE_THRESHOLD is shown in Figure 5-53 and described in Table 5-54.

Figure 5-53. EUR_CR_DPM_TA_PAGE_THRESHOLD Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-54. EUR_CR_DPM_TA_PAGE_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	Reserved	R/W	00000h	
13-0	VALUE	R/W	0000h	Effective on Loading the TA Page Table Context, the maximum number of pages to allocate before signaling almost out of memory.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.54 EUR_CR_DPM_ZLS_PAGE_THRESHOLD Register (offset = 624h)

EUR_CR_DPM_ZLS_PAGE_THRESHOLD is shown in Figure 5-54 and described in Table 5-55.

Figure 5-54. EUR_CR_DPM_ZLS_PAGE_THRESHOLD Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-55. EUR_CR_DPM_ZLS_PAGE_THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	Reserved	R/W	00000h	
13-0	VALUE	R/W	0000h	Effective on Loading the TA Page Table Context, the maximum number of pages to allocate before signaling out of memory to the host. The ZLS threshold should be set such that sufficient pages are left available to the ZLS module when it performs its dynamic memory allocation upon a partial render. The number of pages spare should be set to be larger than the maximum pages for 1 macro tile of Z data. For example: If 4 pages are required for 1 macro tile of Z, and there are 50 pages in the system, then this threshold should be set to 50 - 4 = 46.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.55 EUR_CR_DPM_TA_GLOBAL_LIST Register (offset = 628h)

EUR_CR_DPM_TA_GLOBAL_LIST is shown in Figure 5-55 and described in Table 5-56.

Figure 5-55. EUR_CR_DPM_TA_GLOBAL_LIST Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-56. EUR_CR_DPM_TA_GLOBAL_LIST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	Reserved	R/W	0000h	
16	POLICY	R/W	0h	Effective on Loading the TA Page Table Context, a value of 1 in this field enables the global list size limit
15-14	Reserved	R/W	0h	
13-0	SIZE	R/W	0000h	Effective on Loading the TA Page Table Context, the maximum number of pages to allocate to the global list before signaling global list out of memory.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.56 EUR_CR_DPM_STATE_TABLE_BASE Register (offset = 62Ch)

Effective immediately, this register defines the base address of the state table when it is read and written to/from memory. The state table itself contains the head, tail and number of pages allocated to every macrotile, and the global list. Each context needs a separate state page table.

EUR_CR_DPM_STATE_TABLE_BASE is shown in Figure 5-56 and described in Table 5-57.

Figure 5-56. EUR_CR_DPM_STATE_TABLE_BASE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-57. EUR_CR_DPM_STATE_TABLE_BASE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved	R/W	0h	
27-4	ADDR	R/W	000000h	256MB Addressable, 16Byte aligned Base Address for State Table in the DPM module.
3-0	Reserved	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.57 EUR_CR_DPM_STATE_CONTEXT_ID Register (offset = 630h)

EUR_CR_DPM_STATE_CONTEXT_ID is shown in Figure 5-57 and described in Table 5-58.

Figure 5-57. EUR_CR_DPM_STATE_CONTEXT_ID Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-58. EUR_CR_DPM_STATE_CONTEXT_ID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	Reserved	R/W	00000000h	
2	ALLOC	R/W	0h	Effective immediately, this field defines the which of the internal contexts is to be used for allocating memory
1	DALLOC	R/W	0h	Effective immediately, this field defines the which of the internal contexts is to be used for de-allocating memory
0	LS	R/W	0h	Effective immediately, this field defines which of the internal contexts is to be loaded or stored from/to memory

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.58 EUR_CR_DPM_CONTROL_TABLE_BASE Register (offset = 634h)

Effective immediately, this register defines the base address of the control table when it is read and written to/from memory. The control table itself contains the current offset into the page, and whether a page has been allocated, for each macrotile. Each context needs a separate state page table.

EUR_CR_DPM_CONTROL_TABLE_BASE is shown in Figure 5-58 and described in Table 5-59.

Figure 5-58. EUR_CR_DPM_CONTROL_TABLE_BASE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-59. EUR_CR_DPM_CONTROL_TABLE_BASE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved	R/W	0h	
27-4	ADDR	R/W	000000h	256MB Addressable, 16Byte aligned Base Address for the Control Table in the DPM module.
3-0	Reserved	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.59 EUR_CR_DPM_START_OF_CONTEXT_PAGES_ALLOCATED Register (offset = 638h)

EUR_CR_DPM_START_OF_CONTEXT_PAGES_ALLOCATED is shown in Figure 5-59 and described in Table 5-60.

Figure 5-59. EUR_CR_DPM_START_OF_CONTEXT_PAGES_ALLOCATED Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-60. EUR_CR_DPM_START_OF_CONTEXT_PAGES_ALLOCATED Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	Reserved	R/W	0h	
29-16	GLOBAL	R/W	0000h	Effective on the loading of the TA Page Table Context, this register defines the number of the pages which are currently allocated to the context global list. The value for this register is read from the matching status register, on a context store, and loaded on a context load.
15-14	Reserved	R/W	0h	
13-0	LOCAL	R/W	0000h	Effective on the loading of the TA Page Table Context, this register defines the number of the pages which are currently allocated to the context. The value for this register is read from the matching status register, on a context store, and loaded on a context load.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.60 EUR_CR_DPM_3D_DEALLOCATE Register (offset = 63Ch)

EUR_CR_DPM_3D_DEALLOCATE is shown in Figure 5-60 and described in Table 5-61.

Figure 5-60. EUR_CR_DPM_3D_DEALLOCATE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-61. EUR_CR_DPM_3D_DEALLOCATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	Reserved	R/W	00000000h	
1	ENABLE	R/W	0h	Effective on start render, this field indicates that the hardware should de-allocate Effective on start render, this field indicates that the hardware should de-allocate memory.mory.
0	GLOBAL	R/W	0h	Effective on start render, this field indicates that the hardware should de-allocate the memory associated with the global list if it is set.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.61 EUR_CR_CORE_ID Register (offset = 10h)

Core ID Register

EUR_CR_CORE_ID is shown in Figure 5-61 and described in Table 5-62.

Figure 5-61. EUR_CR_CORE_ID Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-62. EUR_CR_CORE_ID Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	ID	R	0112h	Core ID
15-0	CONFIG	R	0000h	Core Configuration

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.62 EUR_CR_DPM_ALLOC Register (offset = 640h)

A write to this register invalidates the current contents and causes a new allocation cycle

EUR_CR_DPM_ALLOC is shown in Figure 5-62 and described in Table 5-63.

Figure 5-62. EUR_CR_DPM_ALLOC Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-63. EUR_CR_DPM_ALLOC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R/W	0000h	
17	PAGE_OUTOFMEMORY	R/W	0h	Out of memory. Indicates that the action is pending due to an out of memory condition.
16	PAGE_VALID	R/W	0h	On a read a value of 1 in this field indicates that the page has been allocated, and that the page is valid.
15-14	Reserved	R/W	0h	
13-0	PAGE	R/W	0000h	A read of this register returns the page allocated to the host, when valid. If it is not valid the return data is undefined.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.63 EUR_CR_DPM_DALLOC Register (offset = 644h)

A write to this register invalidates the current contents and causes a new de-allocation cycle

EUR_CR_DPM_DALLOC is shown in Figure 5-63 and described in Table 5-64.

Figure 5-63. EUR_CR_DPM_DALLOC Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-64. EUR_CR_DPM_DALLOC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	Reserved	R/W	0000h	
16	PAGE_FREE	R/W	0h	A read indicates whether the page has been freed
15-14	Reserved	R/W	0h	
13-0	PAGE	R/W	FFFFh	The page to be de-allocated by the hardware

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.64 EUR_CR_DPM_TA_ALLOC Register (offset = 648h)

EUR_CR_DPM_TA_ALLOC is shown in Figure 5-64 and described in Table 5-65.

Figure 5-64. EUR_CR_DPM_TA_ALLOC Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-65. EUR_CR_DPM_TA_ALLOC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	Reserved	R/W	0h	
29-16	FREE_LIST_PREVIOUS	R/W	0000h	Effective on Load TA Page Table Context: TA Alloc Free List Previous Page. The DPM maintains the previous and current free list pointers. On a context load this value must be loaded in. For the initial case it should be set to the penultimate page after initialisation (normally TAIL - 1).
15-0	Reserved	R/W	0000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.65 EUR_CR_DPM_3D Register (offset = 64Ch)

EUR_CR_DPM_3D is shown in Figure 5-65 and described in Table 5-66.

Figure 5-65. EUR_CR_DPM_3D Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-66. EUR_CR_DPM_3D Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	Reserved	R/W	0h	
29-16	FREE_LIST_PREVIOUS	R/W	0000h	Effective on Load 3D Page Table Context: 3D Free List Previous Page. The DPM maintains the previous and current free list pointers. On a context load this value must be loaded in. For the initial case it should be set to the penultimate page after initialisation (normally TAIL - 1).
15-0	Reserved	R/W	0000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.66 EUR_CR_DPM_PARTIAL_RENDER Register (offset = 658h)

EUR_CR_DPM_PARTIAL_RENDER is shown in Figure 5-66 and described in Table 5-67.

Figure 5-66. EUR_CR_DPM_PARTIAL_RENDER Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-67. EUR_CR_DPM_PARTIAL_RENDER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R/W	00000000h	
0	ENABLE	R/W	0h	Once The TA DPM has run out of memory, and the driver has scheduled an abort, there must be a partial render performed on the context in order to free memory. This bit is set before the partial render is started.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.67 EUR_CR_DPM_LSS_PARTIAL_CONTEXT Register (offset = 65Ch)

EUR_CR_DPM_LSS_PARTIAL_CONTEXT is shown in Figure 5-67 and described in Table 5-68.

Figure 5-67. EUR_CR_DPM_LSS_PARTIAL_CONTEXT Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-68. EUR_CR_DPM_LSS_PARTIAL_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R/W	00000000h	
0	OPERATION	R/W	0h	When set indicates that on a Context state table load, reset or store only the best macro tile, rather than the full scene.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.68 EUR_CR_DPM_START_OF_3D_CONTEXT_PAGES_ALLOCATED Register (offset = 660h)

EUR_CR_DPM_START_OF_3D_CONTEXT_PAGES_ALLOCATED is shown in Figure 5-68 and described in Table 5-69.

Figure 5-68. EUR_CR_DPM_START_OF_3D_CONTEXT_PAGES_ALLOCATED Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-69. EUR_CR_DPM_START_OF_3D_CONTEXT_PAGES_ALLOCATED Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	Reserved	R/W	0h	
29-16	GLOBAL	R/W	0000h	Effective on the loading of the 3D Page Table Context, this register defines the number of the pages which are currently allocated to the context global list. The value for this register is read from the matching status register, on a context store, and loaded on a context load.
15-14	Reserved	R/W	0h	
13-0	LOCAL	R/W	0000h	Effective on the loading of the 3D Page Table Context, this register defines the number of the pages which are currently allocated to the context. The value for this register is read from the matching status register, on a context store, and loaded on a context load.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.69 EUR_CR_DPM_TASK_3D_FREE Register (offset = 680h)

EUR_CR_DPM_TASK_3D_FREE is shown in Figure 5-69 and described in Table 5-70.

Figure 5-69. EUR_CR_DPM_TASK_3D_FREE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-70. EUR_CR_DPM_TASK_3D_FREE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	W	00000000h	
0	LOAD	W	0h	Causes the 3D free list context to be loaded from the relevant configuration registers

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.70 EUR_CR_DPM_TASK_TA_FREE Register (offset = 684h)

EUR_CR_DPM_TASK_TA_FREE is shown in Figure 5-70 and described in Table 5-71.

Figure 5-70. EUR_CR_DPM_TASK_TA_FREE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-71. EUR_CR_DPM_TASK_TA_FREE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	W	00000000h	
0	LOAD	W	0h	Causes the TA free list context to be loaded from the relevant configuration registers

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.71 EUR_CR_DPM_TASK_STATE Register (offset = 694h)

This register controls the DPM module task state for page tables. The normal order of operation is Store context, Clear context, Load new context.

EUR_CR_DPM_TASK_STATE is shown in Figure 5-71 and described in Table 5-72.

Figure 5-71. EUR_CR_DPM_TASK_STATE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-72. EUR_CR_DPM_TASK_STATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	Reserved	W	00000000h	
2	LOAD	W	0h	Causes the state for the DPM page table to be read from memory. Only the selected context ID is loaded.
1	CLEAR	W	0h	Causes the state for the DPM page table to be cleared. Only the selected context ID is cleared.
0	STORE	W	0h	Causes the state for the DPM page table (non-destructive - the state is not cleared as it is written) to be written to memory. Only the selected context ID is written out.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.72 EUR_CR_DPM_TASK_CONTROL Register (offset = 698h)

This register controls the DPM module task state for control page tables. The normal order of operation is Store context, Clear context, Load new context.

EUR_CR_DPM_TASK_CONTROL is shown in Figure 5-72 and described in Table 5-73.

Figure 5-72. EUR_CR_DPM_TASK_CONTROL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-73. EUR_CR_DPM_TASK_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	Reserved	W	00000000h	
2	LOAD	W	0h	Causes the state for the control page table to be read from memory.
1	CLEAR	W	0h	Causes the state for the control page table to be cleared.
0	STORE	W	0h	Causes the state for the control table (non-destructive - the control is not cleared as it is written) to be written to memory.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.73 EUR_CR_DPM_OUTOFMEM Register (offset = 69Ch)

EUR_CR_DPM_OUTOFMEM is shown in Figure 5-73 and described in Table 5-74.

Figure 5-73. EUR_CR_DPM_OUTOFMEM Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-74. EUR_CR_DPM_OUTOFMEM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	Reserved	W	00000000h	
2	ABORTALL	W	0h	Causes the hardware to initiate the abort cycle, resulting in an abort complete, and requiring a subsequent full render, in the order of busiest to least busy macro tile. The region headers need to be re-ordered by the driver to guarantee this condition
1	ABORT	W	0h	Causes the hardware to initiate the abort cycle, resulting in an abort complete, and requiring a subsequent partial render
0	RESTART	W	0h	Causes the parameter management to resume trying to allocate pages

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.74 EUR_CR_DPM_FREE_CONTEXT Register (offset = 6A0h)

EUR_CR_DPM_FREE_CONTEXT is shown in Figure 5-74 and described in Table 5-75.

Figure 5-74. EUR_CR_DPM_FREE_CONTEXT Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-75. EUR_CR_DPM_FREE_CONTEXT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	W	00000000h	
0	NOW	W	0h	Causes the hardware to free the memory associated with the current TA context

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.75 EUR_CR_DPM_3D_TIMEOUT Register (offset = 6A4h)

EUR_CR_DPM_3D_TIMEOUT is shown in Figure 5-75 and described in Table 5-76.

Figure 5-75. EUR_CR_DPM_3D_TIMEOUT Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-76. EUR_CR_DPM_3D_TIMEOUT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	W	00000000h	
0	NOW	W	0h	Causes the hardware to free the memory associated with the current 3D context, and if a render is in progress, treat this as if it were complete.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.76 EUR_CR_DPM_TA_EVM Register (offset = 6A8h)

EUR_CR_DPM_TA_EVM is shown in Figure 5-76 and described in Table 5-77.

Figure 5-76. EUR_CR_DPM_TA_EVM Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-77. EUR_CR_DPM_TA_EVM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	W	00000000h	
0	INIT	W	0h	Initialise the TA linked list (between start and end only), to an area of linear memory

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.77 EUR_CR_DPM_TA_ALLOC_FREE_LIST_STATUS1 Register (offset = 700h)

This status register is read for a context switch, and reprogrammed on a context load.

EUR_CR_DPM_TA_ALLOC_FREE_LIST_STATUS1 is shown in Figure 5-77 and described in Table 5-78.

Figure 5-77. EUR_CR_DPM_TA_ALLOC_FREE_LIST_STATUS1 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-78. EUR_CR_DPM_TA_ALLOC_FREE_LIST_STATUS1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	Reserved	R	0h	
29-16	TAIL	R	0000h	The current tail of the free list for the TA allocation context
15-14	Reserved	R	0h	
13-0	HEAD	R	0000h	The current head of the free list for the TA allocation context

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.78 EUR_CR_DPM_3D_FREE_LIST_STATUS1 Register (offset = 70Ch)

This status register is read for a context switch, and reprogrammed on a context load.

EUR_CR_DPM_3D_FREE_LIST_STATUS1 is shown in Figure 5-78 and described in Table 5-79.

Figure 5-78. EUR_CR_DPM_3D_FREE_LIST_STATUS1 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-79. EUR_CR_DPM_3D_FREE_LIST_STATUS1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	Reserved	R	0h	
29-16	TAIL	R	0000h	The current tail of the free list for the 3D context
15-14	Reserved	R	0h	
13-0	HEAD	R	0000h	The current head of the free list for the 3D context

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.79 EUR_CR_DPM_TA_ALLOC_FREE_LIST_STATUS2 Register (offset = 710h)

This status register is read for a context switch, and reprogrammed on a context load.

EUR_CR_DPM_TA_ALLOC_FREE_LIST_STATUS2 is shown in Figure 5-79 and described in Table 5-80.

Figure 5-79. EUR_CR_DPM_TA_ALLOC_FREE_LIST_STATUS2 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-80. EUR_CR_DPM_TA_ALLOC_FREE_LIST_STATUS2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	Reserved	R	00000h	
13-0	PREVIOUS	R	0000h	The element before the tail of the free list for the TA allocation context

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.80 EUR_CR_DPM_3D_FREE_LIST_STATUS2 Register (offset = 71Ch)

This status register is read for a context switch, and reprogrammed on a context load.

EUR_CR_DPM_3D_FREE_LIST_STATUS2 is shown in Figure 5-80 and described in Table 5-81.

Figure 5-80. EUR_CR_DPM_3D_FREE_LIST_STATUS2 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-81. EUR_CR_DPM_3D_FREE_LIST_STATUS2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	Reserved	R	00000h	
13-0	PREVIOUS	R	0000h	The element before the tail of the free list for the 3D context

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.81 EUR_CR_DPM_ABORT_STATUS_MTILE Register (offset = 720h)

This status register is read for a context switch, and reprogrammed on a context load.

EUR_CR_DPM_ABORT_STATUS_MTILE is shown in Figure 5-81 and described in Table 5-82.

Figure 5-81. EUR_CR_DPM_ABORT_STATUS_MTILE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-82. EUR_CR_DPM_ABORT_STATUS_MTILE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	Reserved	R	0000000h	
4	GLOBAL	R	0h	Indicates that the whole scene must be aborted
3-0	INDEX	R	0h	The macro tile which will be aborted by the hardware

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.82 EUR_CR_DPM_PAGE_STATUS Register (offset = 724h)

This status register is read for a context switch, and reprogrammed on a context load.

EUR_CR_DPM_PAGE_STATUS is shown in Figure 5-82 and described in Table 5-83.

Figure 5-82. EUR_CR_DPM_PAGE_STATUS Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-83. EUR_CR_DPM_PAGE_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	Reserved	R	0h	
29-16	TA	R	0000h	The number of pages allocated by the hardware, for the TA context
15-14	Reserved	R	0h	
13-0	TOTAL	R	0000h	The total number of pages allocated by the hardware, across all contexts

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.83 EUR_CR_DPM_PAGE Register (offset = 728h)

This status register is read for a context switch, and reprogrammed on a context load.

EUR_CR_DPM_PAGE is shown in Figure 5-83 and described in Table 5-84.

Figure 5-83. EUR_CR_DPM_PAGE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-84. EUR_CR_DPM_PAGE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	Reserved	R	00000h	
13-0	STATUS_3D	R	0000h	The number of pages allocated by the hardware, for the 3D context

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.84 EUR_CR_DPM_GLOBAL_PAGE_STATUS Register (offset = 72Ch)

This status register is read for a context switch, and reprogrammed on a context load.

EUR_CR_DPM_GLOBAL_PAGE_STATUS is shown in Figure 5-84 and described in Table 5-85.

Figure 5-84. EUR_CR_DPM_GLOBAL_PAGE_STATUS Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-85. EUR_CR_DPM_GLOBAL_PAGE_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	Reserved	R	0h	
29-16	TA	R	0000h	The number of global pages allocated by the hardware, for the TA context
15-14	Reserved	R	0h	
13-0	TOTAL	R	0000h	The total number of global pages allocated by the hardware, across all contexts

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.85 EUR_CR_DPM_GLOBAL_PAGE Register (offset = 730h)

This status register is read for a context switch, and reprogrammed on a context load.

EUR_CR_DPM_GLOBAL_PAGE is shown in Figure 5-85 and described in Table 5-86.

Figure 5-85. EUR_CR_DPM_GLOBAL_PAGE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-86. EUR_CR_DPM_GLOBAL_PAGE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-14	Reserved	R	00000h	
13-0	STATUS_3D	R	0000h	The total number of global pages allocated by the hardware, across the 3D contexts

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.86 EUR_CR_DPM_REQUESTING Register (offset = 734h)

EUR_CR_DPM_REQUESTING is shown in Figure 5-86 and described in Table 5-87.

Figure 5-86. EUR_CR_DPM_REQUESTING Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-87. EUR_CR_DPM_REQUESTING Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	Reserved	R	00000000h	
1-0	SOURCE	R	0h	The device that is requesting memory; Bit 0 is the TA (Control Stream Request), Bit 1 is the TA (Vertex Data Stream Request)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.87 EUR_CR_DPM_RNG_HDR_PARSER Register (offset = 738h)

EUR_CR_DPM_RNG_HDR_PARSER is shown in Figure 5-87 and described in Table 5-88.

Figure 5-87. EUR_CR_DPM_RNG_HDR_PARSER Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-88. EUR_CR_DPM_RNG_HDR_PARSER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R	0h	
0	IDLE	R	1h	The DPM Region Header IDLE indication, 1 = idle, 0 = busy

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.88 EUR_CR_CORE_REVISION Register (offset = 14h)

Core Revision Register identifies the specific core revision. This is updated to reflect the formal release status of the core.

EUR_CR_CORE_REVISION is shown in Figure 5-88 and described in Table 5-89.

Figure 5-88. EUR_CR_CORE_REVISION Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-89. EUR_CR_CORE_REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	DESIGNER	R	00h	Designer Field
23-16	MAJOR	R	01h	Major Revision
15-8	MINOR	R	02h	Minor Revision
7-0	MAINTENANCE	R	0Fh	Maintenance Revision

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.89 EUR_CR_TSP_CONFIG Register (offset = 800h)

EUR_CR_TSP_CONFIG is shown in Figure 5-89 and described in Table 5-90.

Figure 5-89. EUR_CR_TSP_CONFIG Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-90. EUR_CR_TSP_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	Reserved	R/W	00000000h	
2	CEMEDGE_DONTFILTER	R/W	0h	Disable filtering over edges/corners for CEM
1	CEMGRAD_DONTNEGATE	R/W	0h	Disable negation for user supplied gradients for cem swap i.e. will only swap dudx etc not negate
0	Reserved	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.90 EUR_CR_CACHE_CTRL Register (offset = 804h)

Unified Cache global control register

EUR_CR_CACHE_CTRL is shown in Figure 5-90 and described in Table 5-91.

Figure 5-90. EUR_CR_CACHE_CTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-91. EUR_CR_CACHE_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	Reserved	R/W	0h	
29	L0P0OFF	R/W	0h	Cache L0 Pipe1 Bypass
28	INVALIDATE	R/W	0h	Cache Invalidate - To invalidate the cache, ie remove all contents from the cache, then setting this bit to 1 will start an invalidation cycle within the cache. Immediately the cache will not except anymore read request until the invalidation cycle completes. All requests pending in the cache (L0/L1/L2) will be completed before the cache completes its invalidation cycle. On completion of the invalidation cycle an interrupt will occur, if enabled, and the MADD_CACHE_INVALCOMP bit will be set and the cache will return to its normal operation
27	Reserved	R/W	0h	
26	L1P1OFF	R/W	0h	Cache Off L1 Pipe1 - When set to 1 turns off the L1 cache associated with Pipe 1. Read request to the L1 cache will be forwarded directly to the L2 cache. The cache must be invalidated after the bit is set to 0 (Cache Level On)
25	L1P0OFF	R/W	0h	Cache Off L1 Pipe0 - When set to 1 turns off the L1 cache associated with Pipe 0. Read request to the L1 cache will be forwarded directly to the L2 cache. The cache must be invalidated after the bit is set to 0 (Cache Level On)
24-16	Reserved	R/W	000h	
15-12	PARTDM3	R/W	0h	Cache Partition Mask for DM3 - defines which cache lines are used for Data Master type 3. Each bit represents one cache line, where there are 4 cache lines in the cache. Each cache line contains 8 possible address tags which are compared against the incoming address to determine if the data requested is in the cache or not. If all bits are set then there are 4 x 8 possible address tags i.e. DM3 read request will try and match against all the cache ram contents. Setting a bit to 0, will exclude that cache line from being compared against the incoming request address
11-8	PARTDM2	R/W	0h	Cache Partition Mask for DM2 - Refer to PARTDM3, but with respect to DM2
7-4	PARTDM1	R/W	0h	Cache Partition Mask for DM1 - Refer to PARTDM3, but with respect to DM1
3-0	PARTDM0	R/W	0h	Cache Partition Mask for DM0 - Refer to PARTDM3, but with respect to DM0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.91 EUR_CR_DESIGNER_REV_FIELD1 Register (offset = 18h)

Designer Revision Field. The SOC designer can use this register for their own revision control if required.

EUR_CR_DESIGNER_REV_FIELD1 is shown in Figure 5-91 and described in Table 5-92.

Figure 5-91. EUR_CR_DESIGNER_REV_FIELD1 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-92. EUR_CR_DESIGNER_REV_FIELD1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DESIGNER_REV_FIELD1	R	00000000h	Designer Field

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.92 EUR_CR_TRIGGER Register (offset = 100h)

TA/3D Lockup counter timeout value (16K cycle granularity) (Debug only, for further information contact Imagination Technologies)

EUR_CR_TRIGGER is shown in Figure 5-92 and described in Table 5-93.

Figure 5-92. EUR_CR_TRIGGER Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-93. EUR_CR_TRIGGER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	Reserved	R/W	0h	
4-0	TRIGGER_COUNT	R/W	03h	Trigger count value in 16K cycles for TA and 3D lockup triggers

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.93 EUR_CR_USE_CTRL Register (offset = A00h)

EUR_CR_USE_CTRL is shown in Figure 5-93 and described in Table 5-94.

Figure 5-93. EUR_CR_USE_CTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-94. EUR_CR_USE_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	Reserved	R/W	000h	
19	REGBOUND_ZERO	R/W	0h	0: RETURN_ONE Controls whether out-of-range indexed-mode MOE register numbers redirect to 0 for alpha (when set) or 1 (when cleared) 0 RETURN_ONE 1 RETURN_ZERO
18-14	INSTLIMIT	R/W	00h	Maximum number of times an active slot can be serviced in a row
13-4	Reserved	R/W	000h	
3-2	KILL_MODE	R/W	0h	0x0 Off 1 - Flush: Flush 2 - Kill: Kill 3 - Reserved_3: Reserved
1-0	KILL_DM	R/W	0h	Data master to kill

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.94 EUR_CR_USE_LD_REDIRECT Register (offset = A04h)

EUR_CR_USE_LD_REDIRECT is shown in Figure 5-94 and described in Table 5-95.

Figure 5-94. EUR_CR_USE_LD_REDIRECT Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-95. EUR_CR_USE_LD_REDIRECT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	ADDR	R/W	0000000h	Byte-aligned memory address to redirect LD instructions which fail range check
3-0	Reserved	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.95 EUR_CR_USE_CACHE Register (offset = A08h)

EUR_CR_USE_CACHE is shown in Figure 5-95 and described in Table 5-96.

Figure 5-95. EUR_CR_USE_CACHE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-96. EUR_CR_USE_CACHE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	W	0h	
0	INVALIDATE	W	0h	Any write to this location invalidates the instruction cache automatically

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.96 EUR_CR_USE_CODE_BASE00 Register (offset = A0Ch)

EUR_CR_USE_CODE_BASE00 is shown in Figure 5-96 and described in Table 5-97.

Figure 5-96. EUR_CR_USE_CODE_BASE00 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-97. EUR_CR_USE_CODE_BASE00 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	W	0h	
25-24	DM	W	0h	Data Master
23-0	ADDR	W	000000h	Code Base Address

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.97 EUR_CR_USE_CODE_BASE01 Register (offset = A10h)

EUR_CR_USE_CODE_BASE01 is shown in Figure 5-97 and described in Table 5-98.

Figure 5-97. EUR_CR_USE_CODE_BASE01 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-98. EUR_CR_USE_CODE_BASE01 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	W	0h	
25-24	DM	W	0h	Data Master
23-0	ADDR	W	000000h	Code Base Address

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.98 EUR_CR_USE_CODE_BASE02 Register (offset = A14h)

EUR_CR_USE_CODE_BASE02 is shown in Figure 5-98 and described in Table 5-99.

Figure 5-98. EUR_CR_USE_CODE_BASE02 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-99. EUR_CR_USE_CODE_BASE02 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	W	0h	
25-24	DM	W	0h	Data Master
23-0	ADDR	W	000000h	Code Base Address

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.99 EUR_CR_USE_CODE_BASE03 Register (offset = A18h)

EUR_CR_USE_CODE_BASE03 is shown in Figure 5-99 and described in Table 5-100.

Figure 5-99. EUR_CR_USE_CODE_BASE03 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-100. EUR_CR_USE_CODE_BASE03 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	W	0h	
25-24	DM	W	0h	Data Master
23-0	ADDR	W	000000h	Code Base Address

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.100 EUR_CR_USE_CODE_BASE04 Register (offset = A1Ch)

EUR_CR_USE_CODE_BASE04 is shown in Figure 5-100 and described in Table 5-101.

Figure 5-100. EUR_CR_USE_CODE_BASE04 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-101. EUR_CR_USE_CODE_BASE04 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	W	0h	
25-24	DM	W	0h	Data Master
23-0	ADDR	W	000000h	Code Base Address

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.101 EUR_CR_USE_CODE_BASE05 Register (offset = A20h)

EUR_CR_USE_CODE_BASE05 is shown in Figure 5-101 and described in Table 5-102.

Figure 5-101. EUR_CR_USE_CODE_BASE05 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-102. EUR_CR_USE_CODE_BASE05 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	W	0h	
25-24	DM	W	0h	Data Master
23-0	ADDR	W	000000h	Code Base Address

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.102 EUR_CR_USE_CODE_BASE06 Register (offset = A24h)

EUR_CR_USE_CODE_BASE06 is shown in Figure 5-102 and described in Table 5-103.

Figure 5-102. EUR_CR_USE_CODE_BASE06 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-103. EUR_CR_USE_CODE_BASE06 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	W	0h	
25-24	DM	W	0h	Data Master
23-0	ADDR	W	000000h	Code Base Address

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.103 EUR_CR_USE_CODE_BASE07 Register (offset = A28h)

EUR_CR_USE_CODE_BASE07 is shown in Figure 5-103 and described in Table 5-104.

Figure 5-103. EUR_CR_USE_CODE_BASE07 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-104. EUR_CR_USE_CODE_BASE07 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	W	0h	
25-24	DM	W	0h	Data Master
23-0	ADDR	W	000000h	Code Base Address

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.104 EUR_CR_USE_CODE_BASE08 Register (offset = A2Ch)

EUR_CR_USE_CODE_BASE08 is shown in Figure 5-104 and described in Table 5-105.

Figure 5-104. EUR_CR_USE_CODE_BASE08 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-105. EUR_CR_USE_CODE_BASE08 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	W	0h	
25-24	DM	W	0h	Data Master
23-0	ADDR	W	000000h	Code Base Address

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.105 EUR_CR_USE_CODE_BASE09 Register (offset = A30h)

EUR_CR_USE_CODE_BASE09 is shown in Figure 5-105 and described in Table 5-106.

Figure 5-105. EUR_CR_USE_CODE_BASE09 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-106. EUR_CR_USE_CODE_BASE09 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	W	0h	
25-24	DM	W	0h	Data Master
23-0	ADDR	W	000000h	Code Base Address

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.106 EUR_CR_USE_CODE_BASE10 Register (offset = A34h)

EUR_CR_USE_CODE_BASE10 is shown in Figure 5-106 and described in Table 5-107.

Figure 5-106. EUR_CR_USE_CODE_BASE10 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-107. EUR_CR_USE_CODE_BASE10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	W	0h	
25-24	DM	W	0h	Data Master
23-0	ADDR	W	000000h	Code Base Address

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.107 EUR_CR_USE_CODE_BASE11 Register (offset = A38h)

EUR_CR_USE_CODE_BASE11 is shown in Figure 5-107 and described in Table 5-108.

Figure 5-107. EUR_CR_USE_CODE_BASE11 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-108. EUR_CR_USE_CODE_BASE11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	W	0h	
25-24	DM	W	0h	Data Master
23-0	ADDR	W	000000h	Code Base Address

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.108 EUR_CR_USE_CODE_BASE12 Register (offset = A3Ch)

EUR_CR_USE_CODE_BASE12 is shown in Figure 5-108 and described in Table 5-109.

Figure 5-108. EUR_CR_USE_CODE_BASE12 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-109. EUR_CR_USE_CODE_BASE12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	W	0h	
25-24	DM	W	0h	Data Master
23-0	ADDR	W	000000h	Code Base Address

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.109 EUR_CR_USE_CODE_BASE13 Register (offset = A40h)

EUR_CR_USE_CODE_BASE13 is shown in Figure 5-109 and described in Table 5-110.

Figure 5-109. EUR_CR_USE_CODE_BASE13 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-110. EUR_CR_USE_CODE_BASE13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	W	0h	
25-24	DM	W	0h	Data Master
23-0	ADDR	W	000000h	Code Base Address

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.110 EUR_CR_USE_CODE_BASE14 Register (offset = A44h)

EUR_CR_USE_CODE_BASE14 is shown in Figure 5-110 and described in Table 5-111.

Figure 5-110. EUR_CR_USE_CODE_BASE14 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-111. EUR_CR_USE_CODE_BASE14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	W	0h	
25-24	DM	W	0h	Data Master
23-0	ADDR	W	000000h	Code Base Address

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.111 EUR_CR_USE_CODE_BASE15 Register (offset = A48h)

EUR_CR_USE_CODE_BASE15 is shown in Figure 5-111 and described in Table 5-112.

Figure 5-111. EUR_CR_USE_CODE_BASE15 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-112. EUR_CR_USE_CODE_BASE15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	W	0h	
25-24	DM	W	0h	Data Master
23-0	ADDR	W	000000h	Code Base Address

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.112 EUR_CR_EVENT_PDS_ENABLE2 Register (offset = A50h)

Enables general event bits into the PDS event data master for processing by the PDS Micro Controller

EUR_CR_EVENT_PDS_ENABLE2 is shown in Figure 5-112 and described in Table 5-113.

Figure 5-112. EUR_CR_EVENT_PDS_ENABLE2 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-113. EUR_CR_EVENT_PDS_ENABLE2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	Reserved	R/W	0h	
4	TRIG_TA	R/W	0h	PDS_Enable: Lockup in TA Phase Trigger
3	TRIG_3D	R/W	0h	PDS_Enable: Lockup in 3d Phase Trigger
2	TRIG_DL	R/W	0h	PDS_Enable: Deadlock has occurred after a Partial Render
1	DPM_3D_FREE_LOAD	R/W	0h	PDS_Enable: DPM 3D Task free load complete
0	DPM_TA_FREE_LOAD	R/W	0h	PDS_Enable: DPM TA Task free load complete

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.113 EUR_CR_PDS_CACHE_ENABLE Register (offset = A54h)

Enables PDS cache invalidation events bits into the PDS event data master for processing by the PDS Micro Controller

EUR_CR_PDS_CACHE_ENABLE is shown in Figure 5-113 and described in Table 5-114.

Figure 5-113. EUR_CR_PDS_CACHE_ENABLE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-114. EUR_CR_PDS_CACHE_ENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R/W	000000h	
8	DSC1_INV3	R/W	0h	PDS_Enable: Context ID 3 has been invalidated in PDS Data Store Cache 1
7	DSC1_INV2	R/W	0h	PDS_Enable: Context ID 2 has been invalidated in PDS Data Store Cache 1
6	DSC1_INV1	R/W	0h	PDS_Enable: Context ID 1 has been invalidated in PDS Data Store Cache 1
5	DSC1_INV0	R/W	0h	PDS_Enable: Context ID 0 has been invalidated in PDS Data Store Cache 1
4	DSC0_INV3	R/W	0h	PDS_Enable: Context ID 3 has been invalidated in PDS Data Store Cache 0
3	DSC0_INV2	R/W	0h	PDS_Enable: Context ID 2 has been invalidated in PDS Data Store Cache 0
2	DSC0_INV1	R/W	0h	PDS_Enable: Context ID 1 has been invalidated in PDS Data Store Cache 0
1	DSC0_INV0	R/W	0h	PDS_Enable: Context ID 0 has been invalidated in PDS Data Store Cache 0
0	CSC_INV	R/W	0h	PDS_Enable: PDS Code Store Cache has been invalidated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.114 EUR_CR_EVENT_PDS_ENABLE Register (offset = A58h)

Enables general event bits into the PDS event data master for processing by the PDS Micro Controller

EUR_CR_EVENT_PDS_ENABLE is shown in Figure 5-114 and described in Table 5-115.

Figure 5-114. EUR_CR_EVENT_PDS_ENABLE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-115. EUR_CR_EVENT_PDS_ENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	Reserved	R/W	0h	
29	TIMER	R/W	0h	PDS_Enable: The PDS Event timer interrupt (signalled on timer traversing through 0).
28	TA_DPM_FAULT	R/W	0h	PDS_Enable: The TA Object Passing System has got out of sync, a Terminate Control Word DPM read was met by StartofBlock data
27	Reserved	R/W	0h	
26	MADD_CACHE_INVALCOMPLETE	R/W	0h	PDS_Enable: Invalidate of Madd L0, L1, L2 Cache Complete
25	DPM_OUT_OF_MEMORY_ZLS	R/W	0h	PDS_Enable: Memory Allocation Failed for ZLS
24	DPM_TA_MEM_FREE	R/W	0h	PDS_Enable: All memory associated with TA context has been freed
23	ISP_END_PASS	R/W	0h	PDS_Enable: ISP End pass
22	DPM_INITEND	R/W	0h	PDS_Enable: DPM Initialisation Complete
21	OTPM_LOADED	R/W	0h	PDS_Enable: The OTPM has loaded its state
20	OTPM_INV	R/W	0h	PDS_Enable: The OTPM has invalidated its contents
19	OTPM_FLUSHED	R/W	0h	PDS_Enable: The OTPM has flushed its state
18	PIXELBE_END_RENDER	R/W	0h	PDS_Enable: End of Scene
17	ISP_HALT	R/W	0h	PDS_Enable: ISP Halt breakpoint triggered
16	ISP_VISIBILITY_FAIL	R/W	0h	PDS_Enable: Visibility Test Fail
15	BREAKPOINT	R/W	0h	PDS_Enable: ISP Breakpoint
14	SW_EVENT	R/W	0h	PDS_Enable: Software triggered event
13	TA_FINISHED	R/W	0h	PDS_Enable: TA complete, (Scene Terminated)
12	TA_TERMINATE	R/W	0h	PDS_Enable: The TA terminate complete (issue as part of complex scene) (MacroTile aborted)
11	TPC_CLEAR	R/W	0h	PDS_Enable: The TPC has reset its state
10	TPC_FLUSH	R/W	0h	PDS_Enable: The TPC has stored its state
9	DPM_CONTROL_CLEAR	R/W	0h	PDS_Enable: The reset of the DPM control context has been completed
8	DPM_CONTROL_LOAD	R/W	0h	PDS_Enable: The load of the DPM control context has been completed
7	DPM_CONTROL_STORE	R/W	0h	PDS_Enable: The store of the DPM control context has been completed
6	DPM_STATE_CLEAR	R/W	0h	PDS_Enable: The reset of the DPM state context has been completed
5	DPM_STATE_LOAD	R/W	0h	PDS_Enable: The load of the DPM state context has been completed
4	DPM_STATE_STORE	R/W	0h	PDS_Enable: The store of the DPM state context has been completed
3	DPM_REACHED_MEM_THRESH	R/W	0h	PDS_Enable: DPM Memory allocation has reached the preset memory threshold for total memory allocation
2	DPM_OUT_OF_MEMORY_GBL	R/W	0h	PDS_Enable: DPM Memory allocation failed for the global list
1	DPM_OUT_OF_MEMORY_MT	R/W	0h	PDS_Enable: DPM Memory allocation failed for a macro tile

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 5-115. EUR_CR_EVENT_PDS_ENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	DPM_3D_MEM_FREE	R/W	0h	PDS_Enable: DPM Memory allocation complete for the current render

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.115 EUR_CR_EVENT_PIXEL_PDS_EXEC Register (offset = A5Ch)

EUR_CR_EVENT_PIXEL_PDS_EXEC is shown in Figure 5-115 and described in Table 5-116.

Figure 5-115. EUR_CR_EVENT_PIXEL_PDS_EXEC Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-116. EUR_CR_EVENT_PIXEL_PDS_EXEC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	R/W	00h	
25-4	ADDR	R/W	000000h	PDS Execution Address for pixel tasks (Positioned as a byte address, 128 bit granularity)
3-0	Reserved	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.116 EUR_CR_EVENT_PIXEL_PDS_DATA Register (offset = A60h)

EUR_CR_EVENT_PIXEL_PDS_DATA is shown in Figure 5-116 and described in Table 5-117.

Figure 5-116. EUR_CR_EVENT_PIXEL_PDS_DATA Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-117. EUR_CR_EVENT_PIXEL_PDS_DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	Reserved	R/W	0000000h	
5-0	SIZE	R/W	00h	PDS Data Size in 128 bit words, Pixel Tasks

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.117 EUR_CR_EVENT_PIXEL_PDS_INFO Register (offset = A64h)

EUR_CR_EVENT_PIXEL_PDS_INFO is shown in Figure 5-117 and described in Table 5-118.

Figure 5-117. EUR_CR_EVENT_PIXEL_PDS_INFO Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-118. EUR_CR_EVENT_PIXEL_PDS_INFO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	R/W	00h	
25	USESECEXEC	R/W	0h	Secondary tasks should execute on the USSE
24-22	Reserved	R/W	0h	
21-13	ATTRIBUTE_SIZE	R/W	000h	The attribute size in 128 bit words
12-11	DM	R/W	0h	The attribute size in 128 bit words 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Reserved_2: Reserved 3 - Event: Event
10	PNS	R/W	0h	Allocate primary not secondary memory
9-8	USE_PIPELINE	R/W	0h	Selects the USSE pipeline allocated to the data master 0 - Data: Data master selects the USSE pipeline 1 - PIPE0: Pipe 0 2 - PIPE1: Pipe 1 3 - Pipes: Pipes 0 and 1
7-1	ODS	R/W	00h	The output data stride in 128 bit words
0	SD	R/W	0h	Sequential Dependency bit. All end of tile tasks should set this bit, in order to ensure that preceding pixel tasks complete before the tile is emitted.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.118 EUR_CR_EVENT_OTHER_PDS_EXEC Register (offset = A68h)

EUR_CR_EVENT_OTHER_PDS_EXEC is shown in Figure 5-118 and described in Table 5-119.

Figure 5-118. EUR_CR_EVENT_OTHER_PDS_EXEC Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-119. EUR_CR_EVENT_OTHER_PDS_EXEC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	R/W	00h	
25-4	ADDR	R/W	000000h	PDS Execution Address for pixel tasks (Positioned as a byte address, 128 bit granularity)
3-0	Reserved	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.119 EUR_CR_EVENT_OTHER_PDS_DATA Register (offset = A6Ch)

EUR_CR_EVENT_OTHER_PDS_DATA is shown in Figure 5-119 and described in Table 5-120.

Figure 5-119. EUR_CR_EVENT_OTHER_PDS_DATA Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-120. EUR_CR_EVENT_OTHER_PDS_DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	Reserved	R/W	0000000h	
5-0	SIZE	R/W	00h	PDS Data Size in 128 bit words, Pixel Tasks

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.120 EUR_CR_EVENT_OTHER_PDS_INFO Register (offset = A70h)

EUR_CR_EVENT_OTHER_PDS_INFO is shown in Figure 5-120 and described in Table 5-121.

Figure 5-120. EUR_CR_EVENT_OTHER_PDS_INFO Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-121. EUR_CR_EVENT_OTHER_PDS_INFO Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	R/W	00h	
25	USESECEXEC	R/W	0h	Secondary tasks should execute on the USSE
24-22	OBUFFER_REQD	R/W	0h	The number of output buffers to be allocated
21-13	ATTRIBUTE_SIZE	R/W	000h	The attribute size in 128 bit words
12-11	DM	R/W	0h	Selects the Data Master 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Reserved_2: Reserved 3 - Event: Event
10	PNS	R/W	0h	Allocate primary not secondary memory
9-8	USE_PIPELINE	R/W	0h	Selects the USSE pipeline allocated to the data master 0 - Data: Data master selects the USSE pipeline 1 - PIPE0: Pipe 0 2 - PIPE1: Pipe 1 3 - Pipes: Pipes 0 and 1
7-1	ODS	R/W	00h	The output data stride in 128 bit words
0	SD	R/W	0h	Sequential Dependency bit. All end of tile tasks should set this bit, in order to ensure that preceding pixel tasks complete before the tile is emitted.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.121 EUR_CR_DMS_CTRL Register (offset = A74h)

Control register used to disable individual data masters - '1' - disable data master, '0' - enable data master

EUR_CR_DMS_CTRL is shown in Figure 5-121 and described in Table 5-122.

Figure 5-121. EUR_CR_DMS_CTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-122. EUR_CR_DMS_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved	R/W	0h	
27-25	MAX_NUM_VERTEX_PARTITIONS	R/W	0h	The maximum number of partitions allowed on each USSE for vertex tasks
24-22	MAX_NUM_PIXEL_PARTITIONS	R/W	0h	The maximum number of partitions allowed on each USSE for pixel tasks
21-16	MAX_NUM_EDM_TASKS	R/W	00h	The maximum number of vertex tasks allowed on each USSE, range 0 to 32
15-10	MAX_NUM_VERTEX_TASKS	R/W	00h	The maximum number of vertex tasks allowed on each USSE, range 0 to 32
9-4	MAX_NUM_PIXEL_TASKS	R/W	00h	The maximum number of pixel tasks allowed on each USSE, range 0 to 32
3	DISABLE_DM_LOOPBACK	R/W	0h	Disable Loopback DM
2	DISABLE_DM_EVENT	R/W	0h	Disable Event DM
1	DISABLE_DM_PIXEL	R/W	0h	Disable Pixel DM
0	DISABLE_DM_VERTEX	R/W	0h	Disable Vertex DM

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.122 EUR_CR_DMS_AGE Register (offset = A78h)

EUR_CR_DMS_AGE is shown in Figure 5-122 and described in Table 5-123.

Figure 5-122. EUR_CR_DMS_AGE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-123. EUR_CR_DMS_AGE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	Reserved	R/W	0000h	
16	ENABLE	R/W	0h	Enable the AGE Control mechanism
15-0	VALUE	R/W	0000h	The maximum age for any task before it will become a a priority to be serviced (in clock cycles)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.123 EUR_CR_USE_G0 Register (offset = A7Ch)

EUR_CR_USE_G0 is shown in Figure 5-123 and described in Table 5-124.

Figure 5-123. EUR_CR_USE_G0 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-124. EUR_CR_USE_G0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	Reserved	R/W	000000h	
7-0	P	R/W	00h	General-purpose, cross-thread-accessible read/write register for USSE programs via the 'special' bank type

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.124 EUR_CR_USE_G1 Register (offset = A80h)

EUR_CR_USE_G1 is shown in Figure 5-124 and described in Table 5-125.

Figure 5-124. EUR_CR_USE_G1 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-125. EUR_CR_USE_G1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	Reserved	R/W	000000h	
7-0	P	R/W	00h	General-purpose, cross-thread-accessible read/write register for USSE programs via the 'special' bank type

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.125 EUR_CR_USE_FILTER0_LEFT Register (offset = A84h)

EUR_CR_USE_FILTER0_LEFT is shown in Figure 5-125 and described in Table 5-126.

Figure 5-125. EUR_CR_USE_FILTER0_LEFT Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-126. EUR_CR_USE_FILTER0_LEFT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	TAP0	R/W	00h	FIRH instruction filter option, signed integer
23-16	TAP1	R/W	00h	FIRH instruction filter option, signed integer
15-8	TAP2	R/W	00h	FIRH instruction filter option, signed integer
7-0	TAP3	R/W	00h	FIRH instruction filter option, signed integer

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.126 EUR_CR_USE_FILTER0_RIGHT Register (offset = A88h)

EUR_CR_USE_FILTER0_RIGHT is shown in Figure 5-126 and described in Table 5-127.

Figure 5-126. EUR_CR_USE_FILTER0_RIGHT Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-127. EUR_CR_USE_FILTER0_RIGHT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	TAP4	R/W	00h	FIRH instruction filter option, signed integer
23-16	TAP5	R/W	00h	FIRH instruction filter option, signed integer
15-8	TAP6	R/W	00h	FIRH instruction filter option, signed integer
7-0	TAP7	R/W	00h	FIRH instruction filter option, signed integer

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.127 EUR_CR_USE_FILTER0_EXTRA Register (offset = A8Ch)

EUR_CR_USE_FILTER0_EXTRA is shown in Figure 5-127 and described in Table 5-128.

Figure 5-127. EUR_CR_USE_FILTER0_EXTRA Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-128. EUR_CR_USE_FILTER0_EXTRA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	Reserved	R/W	000h	
19-4	CONSTANT	R/W	0000h	FIRH instruction filter option, unsigned integer additive constant
3-0	SHR	R/W	0h	FIRH instruction filter option, unsigned integer right shift amount

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.128 EUR_CR_USE_FILTER1_LEFT Register (offset = A90h)

EUR_CR_USE_FILTER1_LEFT is shown in Figure 5-128 and described in Table 5-129.

Figure 5-128. EUR_CR_USE_FILTER1_LEFT Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-129. EUR_CR_USE_FILTER1_LEFT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	TAP0	R/W	00h	FIRH instruction filter option, signed integer
23-16	TAP1	R/W	00h	FIRH instruction filter option, signed integer
15-8	TAP2	R/W	00h	FIRH instruction filter option, signed integer
7-0	TAP3	R/W	00h	FIRH instruction filter option, signed integer

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.129 EUR_CR_USE_FILTER1_RIGHT Register (offset = A94h)

EUR_CR_USE_FILTER1_RIGHT is shown in Figure 5-129 and described in Table 5-130.

Figure 5-129. EUR_CR_USE_FILTER1_RIGHT Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-130. EUR_CR_USE_FILTER1_RIGHT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	TAP4	R/W	00h	FIRH instruction filter option, signed integer
23-16	TAP5	R/W	00h	FIRH instruction filter option, signed integer
15-8	TAP6	R/W	00h	FIRH instruction filter option, signed integer
7-0	TAP7	R/W	00h	FIRH instruction filter option, signed integer

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.130 EUR_CR_USE_FILTER1_EXTRA Register (offset = A98h)

EUR_CR_USE_FILTER1_EXTRA is shown in Figure 5-130 and described in Table 5-131.

Figure 5-130. EUR_CR_USE_FILTER1_EXTRA Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-131. EUR_CR_USE_FILTER1_EXTRA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	Reserved	R/W	000h	
19-4	CONSTANT	R/W	0000h	FIRH instruction filter option, unsigned integer additive constant
3-0	SHR	R/W	0h	FIRH instruction filter option, unsigned integer right shift amount

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.131 EUR_CR_USE_FILTER_TABLE Register (offset = A9Ch)

EUR_CR_USE_FILTER_TABLE is shown in Figure 5-131 and described in Table 5-132.

Figure 5-131. EUR_CR_USE_FILTER_TABLE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-132. EUR_CR_USE_FILTER_TABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	Reserved	R/W	00000000h	
1-0	SEL	R/W	0h	FIRH instruction filter hardwired-table selector

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.132 EUR_CR_EVENT_HOST_ENABLE2 Register (offset = 110h)

This register enables interrupts. Writing a '1' to a bit field enables the relevant Event. All the bit fields correspond exactly to those in the EUR_CR_EVENT_STATUS2 register. The MASTER_INTERRUPT bit is a global enable which overrides the Event enables, i.e. '1' - enable external interrupts, '0' - disable external interrupts.

EUR_CR_EVENT_HOST_ENABLE2 is shown in Figure 5-132 and described in Table 5-133.

Figure 5-132. EUR_CR_EVENT_HOST_ENABLE2 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-133. EUR_CR_EVENT_HOST_ENABLE2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	Reserved	R/W	0000000h	
4	TRIG_TA	R/W	0h	Host_Enable: Lockup in TA Phase Trigger
3	TRIG_3D	R/W	0h	Host_Enable: Lockup in 3d Phase Trigger
2	TRIG_DL	R/W	0h	Host_Enable: Deadlock has occurred after a Partial Render
1	DPM_3D_FREE_LOAD	R/W	0h	Host_Enable: DPM 3D Free Load task complete
0	DPM_TA_FREE_LOAD	R/W	0h	Host_Enable: DPM TA Free Load task complete

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.133 EUR_CR_USE_ST_RANGE Register (offset = AA0h)

EUR_CR_USE_ST_RANGE is shown in Figure 5-133 and described in Table 5-134.

Figure 5-133. EUR_CR_USE_ST_RANGE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-134. EUR_CR_USE_ST_RANGE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VALUE	R/W	00000000h	Range check value for ST instruction

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.134 EUR_CR_USE0_DM_SLOT Register (offset = AA4h)

EUR_CR_USE0_DM_SLOT is shown in Figure 5-134 and described in Table 5-135.

Figure 5-134. EUR_CR_USE0_DM_SLOT Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-135. EUR_CR_USE0_DM_SLOT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SLOT15	R	2h	Pipe 0 Data Master in Slot 15 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event
29-28	SLOT14	R	2h	Pipe 0 Data Master in Slot 14 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event
27-26	SLOT13	R	2h	Pipe 0 Data Master in Slot 13 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event
25-24	SLOT12	R	2h	Pipe 0 Data Master in Slot 12 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event
23-22	SLOT11	R	2h	Pipe 0 Data Master in Slot 11 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event
21-20	SLOT10	R	2h	Pipe 0 Data Master in Slot 10 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event
19-18	SLOT9	R	2h	Pipe 0 Data Master in Slot 9 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event
17-16	SLOT8	R	2h	Pipe 0 Data Master in Slot 8 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event
15-14	SLOT7	R	2h	Pipe 0 Data Master in Slot 7 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event
13-12	SLOT6	R	2h	Pipe 0 Data Master in Slot 6 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event
11-10	SLOT5	R	2h	Pipe 0 Data Master in Slot 5 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 5-135. EUR_CR_USE0_DM_SLOT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9-8	SLOT4	R	2h	Pipe 0 Data Master in Slot 4 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event
7-6	SLOT3	R	2h	Pipe 0 Data Master in Slot 3 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event
5-4	SLOT2	R	2h	Pipe 0 Data Master in Slot 2 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event
3-2	SLOT1	R	2h	Pipe 0 Data Master in Slot 1 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event
1-0	SLOT0	R	2h	Pipe 0 Data Master in Slot 0 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.135 EUR_CR_USE1_DM_SLOT Register (offset = AA8h)

EUR_CR_USE1_DM_SLOT is shown in Figure 5-135 and described in Table 5-136.

Figure 5-135. EUR_CR_USE1_DM_SLOT Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-136. EUR_CR_USE1_DM_SLOT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SLOT15	R	2h	Pipe 0 Data Master in Slot 15 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event
29-28	SLOT14	R	2h	Pipe 0 Data Master in Slot 14 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event
27-26	SLOT13	R	2h	Pipe 0 Data Master in Slot 13 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event
25-24	SLOT12	R	2h	Pipe 0 Data Master in Slot 12 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event
23-22	SLOT11	R	2h	Pipe 0 Data Master in Slot 11 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event
21-20	SLOT10	R	2h	Pipe 0 Data Master in Slot 10 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event
19-18	SLOT9	R	2h	Pipe 0 Data Master in Slot 9 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event
17-16	SLOT8	R	2h	Pipe 0 Data Master in Slot 8 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event
15-14	SLOT7	R	2h	Pipe 0 Data Master in Slot 7 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event
13-12	SLOT6	R	2h	Pipe 0 Data Master in Slot 6 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event
11-10	SLOT5	R	2h	Pipe 0 Data Master in Slot 5 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 5-136. EUR_CR_USE1_DM_SLOT Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9-8	SLOT4	R	2h	Pipe 0 Data Master in Slot 4 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event
7-6	SLOT3	R	2h	Pipe 0 Data Master in Slot 3 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event
5-4	SLOT2	R	2h	Pipe 0 Data Master in Slot 2 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event
3-2	SLOT1	R	2h	Pipe 0 Data Master in Slot 1 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event
1-0	SLOT0	R	2h	Pipe 0 Data Master in Slot 0 0 - Vertex: Vertex 1 - Pixel: Pixel 2 - Empty: Empty 3 - Event: Event

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.136 EUR_CR_USE_EXEF Register (offset = AB4h)

EUR_CR_USE_EXEF is shown in Figure 5-136 and described in Table 5-137.

Figure 5-136. EUR_CR_USE_EXEF Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-137. EUR_CR_USE_EXEF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R/W	0000h	
18-15	CBASE	R/W	0h	Task Flush code baset
14-11	COFFEXT	R/W	0h	Task Flush code offext
10-0	COFF	R/W	000h	Task Flush code offset

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.137 EUR_CR_PDS_EXEC_BASE Register (offset = AB8h)

Base Address in memory where the PDS programs are located

EUR_CR_PDS_EXEC_BASE is shown in Figure 5-137 and described in Table 5-138.

Figure 5-137. EUR_CR_PDS_EXEC_BASE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-138. EUR_CR_PDS_EXEC_BASE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved	R/W	0h	
27-20	ADDR	R/W	00h	256MB Addressable, 1 MB aligned Base Address for PDS programs
19-0	Reserved	R/W	00000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.138 EUR_CR_PDS Register (offset = ABCh)

EUR_CR_PDS is shown in Figure 5-138 and described in Table 5-139.

Figure 5-138. EUR_CR_PDS Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-139. EUR_CR_PDS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	Reserved	R/W	0000000h	
6	DOUT_TIMEOUT_DISABLE	R/W	0h	Disable the internal timeout on PDS code performing necessary DOUT commands
5-0	ATTRIBUTE_CHUNK_START	R/W	00h	The Start Chunk. There are 64 chunks of 32 DWords.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.139 EUR_CR_EVENT_KICKER Register (offset = AC4h)

Base Address in memory where the Kick Synchronisation data will be read from EUR_CR_EVENT_KICKER is shown in Figure 5-139 and described in Table 5-140.

Figure 5-139. EUR_CR_EVENT_KICKER Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-140. EUR_CR_EVENT_KICKER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved	R/W	0h	
27-4	ADDRESS	R/W	000000h	256MB Addressable, 16Byte aligned Base Address for Kick Synchronisation data
3-0	Reserved	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.140 EUR_CR_EVENT_HOST_CLEAR2 Register (offset = 114h)

This register is used to clear event interrupts. Writing a '1' to a bit field clears the relevant Event. All the bit fields correspond exactly to those in the EUR_CR_EVENT_STATUS2 register.

EUR_CR_EVENT_HOST_CLEAR2 is shown in Figure 5-140 and described in Table 5-141.

Figure 5-140. EUR_CR_EVENT_HOST_CLEAR2 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-141. EUR_CR_EVENT_HOST_CLEAR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	Reserved	W	0000000h	
4	TRIG_TA	W	0h	Host_Clear: Lockup in TA Phase Trigger
3	TRIG_3D	W	0h	Host_Clear: Lockup in 3d Phase Trigger
2	TRIG_DL	W	0h	Host_Clear: Deadlock has occurred after a Partial Render
1	DPM_3D_FREE_LOAD	W	0h	Host_Clear: DPM 3D Free Load task complete
0	DPM_TA_FREE_LOAD	W	0h	Host_Clear: DPM TA Free Load task complete

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.141 EUR_CR_EVENT_KICK Register (offset = AC8h)

EUR_CR_EVENT_KICK is shown in Figure 5-141 and described in Table 5-142.

Figure 5-141. EUR_CR_EVENT_KICK Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-142. EUR_CR_EVENT_KICK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	W	00000000h	
0	NOW	W	0h	The kicker register, a write to this register triggers an increment of the PDS kicker.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.142 EUR_CR_EVENT_TIMER Register (offset = ACCh)

The timer which runs at the core clock when enabled, decrementing from the loaded value. Wraps when it hits 0 back to the starting value. A transition through 0 creates an event, which can be enabled into either the PDS or the Host status register. The timer can also be read/written from within the PDS program.

EUR_CR_EVENT_TIMER is shown in Figure 5-142 and described in Table 5-143.

Figure 5-142. EUR_CR_EVENT_TIMER Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-143. EUR_CR_EVENT_TIMER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	Reserved	R/W	00h	
24	ENABLE	R/W	0h	A value of '1' enables the timer, 0 disables the counter (halts without initialisation to 0).
23-0	VALUE	R/W	000000h	The Starting value for the timer (loaded on write from wither PDS or host).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.143 EUR_CR_PDS_INV0 Register (offset = AD0h)

EUR_CR_PDS_INV0 is shown in Figure 5-143 and described in Table 5-144.

Figure 5-143. EUR_CR_PDS_INV0 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-144. EUR_CR_PDS_INV0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	W	00000000h	
0	DSC	W	0h	Invalidate Context 0 in PDS Data Store Caches 0 and 1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.144 EUR_CR_PDS_INV1 Register (offset = AD4h)

EUR_CR_PDS_INV1 is shown in Figure 5-144 and described in Table 5-145.

Figure 5-144. EUR_CR_PDS_INV1 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-145. EUR_CR_PDS_INV1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	W	00000000h	
0	DSC	W	0h	Invalidate Context 1 in PDS Data Store Caches 0 and 1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.145 EUR_CR_PDS_INV2 Register (offset = AD8h)

EUR_CR_PDS_INV2 is shown in Figure 5-145 and described in Table 5-146.

Figure 5-145. EUR_CR_PDS_INV2 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-146. EUR_CR_PDS_INV2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	W	00000000h	
0	DSC	W	0h	Invalidate Context 2 in PDS Data Store Caches 0 and 1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.146 EUR_CR_PDS_INV3 Register (offset = ADCh)

EUR_CR_PDS_INV3 is shown in Figure 5-146 and described in Table 5-147.

Figure 5-146. EUR_CR_PDS_INV3 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-147. EUR_CR_PDS_INV3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	W	00000000h	
0	DSC	W	0h	Invalidate Context 3 in PDS Data Store Caches 0 and 1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.147 EUR_CR_PDS_INV_CSC Register (offset = AE0h)

EUR_CR_PDS_INV_CSC is shown in Figure 5-147 and described in Table 5-148.

Figure 5-147. EUR_CR_PDS_INV_CSC Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-148. EUR_CR_PDS_INV_CSC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	W	00000000h	
0	KICK	W	0h	Invalidate PDS Code Store Cache

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.148 EUR_CR_USE_GLOBCOM0 Register (offset = AE4h)

EUR_CR_USE_GLOBCOM0 is shown in Figure 5-148 and described in Table 5-149.

Figure 5-148. EUR_CR_USE_GLOBCOM0 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-149. EUR_CR_USE_GLOBCOM0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	Reserved	R/W	000000h	
7-0	COUNT	R/W	00h	Counters for cross-thread synchronisation and communication - used for parallelising video processing

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.149 EUR_CR_USE_GLOBCOM1 Register (offset = AE8h)

EUR_CR_USE_GLOBCOM1 is shown in Figure 5-149 and described in Table 5-150.

Figure 5-149. EUR_CR_USE_GLOBCOM1 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-150. EUR_CR_USE_GLOBCOM1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	Reserved	R/W	000000h	
7-0	COUNT	R/W	00h	Counters for cross-thread synchronisation and communication - used for parallelising video processing

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.150 EUR_CR_USE_GLOBCOM2 Register (offset = AECh)

EUR_CR_USE_GLOBCOM2 is shown in Figure 5-150 and described in Table 5-151.

Figure 5-150. EUR_CR_USE_GLOBCOM2 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-151. EUR_CR_USE_GLOBCOM2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	Reserved	R/W	000000h	
7-0	COUNT	R/W	00h	Counters for cross-thread synchronisation and communication - used for parallelising video processing

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.151 EUR_CR_DESIGNER_REV_FIELD2 Register (offset = 1Ch)

Designer Revision Field The SOC designer can use this register for their own revision control if required.

EUR_CR_DESIGNER_REV_FIELD2 is shown in Figure 5-151 and described in Table 5-152.

Figure 5-151. EUR_CR_DESIGNER_REV_FIELD2 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-152. EUR_CR_DESIGNER_REV_FIELD2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DESIGNER_REV_FIELD2	R	00000000h	Designer Field

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.152 EUR_CR_EVENT_STATUS2 Register (offset = 118h)

There are 2 sources of external interrupts in the PowerVR SGX core: General system events and PDS cache invalidation events. The event status2 register indicates the source of any general event interrupt generated by PowerVR SGX. These events only result in an external interrupt if the relevant bit in the EUR_CR_EVENT_HOST_ENABLE register is set. Note1: This register must be combined with the EUR_CR_PDS_CACHE_STATUS register information to determine what has caused the external interrupt. Note2: The host can write to this register in which case any bits written as a '1' are ORed into the register i.e. the host can cause the core to generate an interrupt. (This is typically for debug purposes only)

EUR_CR_EVENT_STATUS2 is shown in Figure 5-152 and described in Table 5-153.

Figure 5-152. EUR_CR_EVENT_STATUS2 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-153. EUR_CR_EVENT_STATUS2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	Reserved	R/W	0000000h	
4	TRIG_TA	R/W	0h	Status: Lockup in TA Phase Trigger
3	TRIG_3D	R/W	0h	Status: Lockup in 3d Phase Trigger
2	TRIG_DL	R/W	0h	Status: Deadlock has occurred after a Partial Render
1	DPM_3D_FREE_LOAD	R/W	0h	Status: DPM 3D free load task complete
0	DPM_TA_FREE_LOAD	R/W	0h	Status: DPM TA free load task complete

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.153 EUR_CR_USE_GLOBCOM3 Register (offset = AF0h)

EUR_CR_USE_GLOBCOM3 is shown in Figure 5-153 and described in Table 5-154.

Figure 5-153. EUR_CR_USE_GLOBCOM3 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-154. EUR_CR_USE_GLOBCOM3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	Reserved	R/W	000000h	
7-0	COUNT	R/W	00h	Counters for cross-thread synchronisation and communication - used for parallelising video processing

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.154 EUR_CR_USE_GLOBCOM4 Register (offset = AF4h)

EUR_CR_USE_GLOBCOM4 is shown in Figure 5-154 and described in Table 5-155.

Figure 5-154. EUR_CR_USE_GLOBCOM4 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-155. EUR_CR_USE_GLOBCOM4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	Reserved	R/W	000000h	
7-0	COUNT	R/W	00h	Counters for cross-thread synchronisation and communication - used for parallelising video processing

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.155 EUR_CR_USE_GLOBCOM5 Register (offset = AF8h)

EUR_CR_USE_GLOBCOM5 is shown in Figure 5-155 and described in Table 5-156.

Figure 5-155. EUR_CR_USE_GLOBCOM5 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-156. EUR_CR_USE_GLOBCOM5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	Reserved	R/W	000000h	
7-0	COUNT	R/W	00h	Counters for cross-thread synchronisation and communication - used for parallelising video processing

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.156 EUR_CR_USE_GLOBCOM6 Register (offset = AFCh)

EUR_CR_USE_GLOBCOM6 is shown in Figure 5-156 and described in Table 5-157.

Figure 5-156. EUR_CR_USE_GLOBCOM6 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-157. EUR_CR_USE_GLOBCOM6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	Reserved	R/W	000000h	
7-0	COUNT	R/W	00h	Counters for cross-thread synchronisation and communication - used for parallelising video processing

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.157 EUR_CR_USE_GLOBCOM7 Register (offset = B00h)

EUR_CR_USE_GLOBCOM7 is shown in Figure 5-157 and described in Table 5-158.

Figure 5-157. EUR_CR_USE_GLOBCOM7 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-158. EUR_CR_USE_GLOBCOM7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	Reserved	R/W	000000h	
7-0	COUNT	R/W	00h	Counters for cross-thread synchronisation and communication - used for parallelising video processing

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.158 EUR_CR_LOOPBACK Register (offset = B04h)

This USE can issue loopback tasks through the PDS back to itself. This bit indicates the status of this interface. A value of 1 should that there are no pending loopback requests (for any USE pipeline), A value of 0 indicates that a loop task is pending, and that further loopback events may cause the USE pipeline to wait until new resource becomes available. Issuing loopback events when tasks are already pending is not recommended, as this may result in system lockups.

EUR_CR_LOOPBACK is shown in Figure 5-158 and described in Table 5-159.

Figure 5-158. EUR_CR_LOOPBACK Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-159. EUR_CR_LOOPBACK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R	00000000h	
0	STATUS	R	1h	The Starting value for the timer (loaded on write from wither PDS or host).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.159 EUR_CR_USE0_SERV_PIXEL Register (offset = B08h)

EUR_CR_USE0_SERV_PIXEL is shown in Figure 5-159 and described in Table 5-160.

Figure 5-159. EUR_CR_USE0_SERV_PIXEL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-160. EUR_CR_USE0_SERV_PIXEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	Reserved	R	0000h	
16	EMPTY	R	1h	No Pixel Data Master tasks in USE0 queue
15-0	COUNT	R	0000h	Number of Pixel Data Master tasks serviced by USE0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.160 EUR_CR_USE0_SERV_VERTEX Register (offset = B0Ch)

EUR_CR_USE0_SERV_VERTEX is shown in Figure 5-160 and described in Table 5-161.

Figure 5-160. EUR_CR_USE0_SERV_VERTEX Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-161. EUR_CR_USE0_SERV_VERTEX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	Reserved	R	0000h	
16	EMPTY	R	1h	No Vertex Data Master tasks in USE0 queue
15-0	COUNT	R	0000h	Number of Vertex Data Master tasks serviced by USE0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.161 EUR_CR_USE0_SERV_EVENT Register (offset = B10h)

EUR_CR_USE0_SERV_EVENT is shown in Figure 5-161 and described in Table 5-162.

Figure 5-161. EUR_CR_USE0_SERV_EVENT Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-162. EUR_CR_USE0_SERV_EVENT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	Reserved	R	0000h	
16	EMPTY	R	1h	No Event Data Master tasks in USE0 queue
15-0	COUNT	R	0000h	Number of Event Data Master tasks serviced by USE0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.162 EUR_CR_USE1_SERV_PIXEL Register (offset = B14h)

EUR_CR_USE1_SERV_PIXEL is shown in Figure 5-162 and described in Table 5-163.

Figure 5-162. EUR_CR_USE1_SERV_PIXEL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-163. EUR_CR_USE1_SERV_PIXEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	Reserved	R	0000h	
16	EMPTY	R	1h	No Pixel Data Master tasks in USE1 queue
15-0	COUNT	R	0000h	Number of Pixel Data Master tasks serviced by USE1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.163 EUR_CR_USE1_SERV_VERTEX Register (offset = B18h)

EUR_CR_USE1_SERV_VERTEX is shown in Figure 5-163 and described in Table 5-164.

Figure 5-163. EUR_CR_USE1_SERV_VERTEX Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-164. EUR_CR_USE1_SERV_VERTEX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	Reserved	R	0000h	
16	EMPTY	R	1h	No Vertex Data Master tasks in USE1 queue
15-0	COUNT	R	0000h	Number of Vertex Data Master tasks serviced by USE1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.164 EUR_CR_USE1_SERV_EVENT Register (offset = B1Ch)

EUR_CR_USE1_SERV_EVENT is shown in Figure 5-164 and described in Table 5-165.

Figure 5-164. EUR_CR_USE1_SERV_EVENT Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-165. EUR_CR_USE1_SERV_EVENT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	Reserved	R	0000h	
16	EMPTY	R	1h	No Event Data Master tasks in USE1 queue
15-0	COUNT	R	0000h	Number of Event Data Master tasks serviced by USE1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.165 EUR_CR_USE_FILTER2_LEFT Register (offset = B20h)

EUR_CR_USE_FILTER2_LEFT is shown in Figure 5-165 and described in Table 5-166.

Figure 5-165. EUR_CR_USE_FILTER2_LEFT Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-166. EUR_CR_USE_FILTER2_LEFT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	TAP0	R/W	00h	FIRH instruction filter option, signed integer
23-8	Reserved	R/W	0000h	
7-0	TAP3	R/W	00h	FIRH instruction filter option, signed integer

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.166 EUR_CR_USE_FILTER2_RIGHT Register (offset = B24h)

EUR_CR_USE_FILTER2_RIGHT is shown in Figure 5-166 and described in Table 5-167.

Figure 5-166. EUR_CR_USE_FILTER2_RIGHT Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-167. EUR_CR_USE_FILTER2_RIGHT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	Reserved	R/W	0000h	
15-8	TAP6	R/W	00h	FIRH instruction filter option, signed integer
7-0	Reserved	R/W	00h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.167 EUR_CR_USE_FILTER2_EXTRA Register (offset = B28h)

EUR_CR_USE_FILTER2_EXTRA is shown in Figure 5-167 and described in Table 5-168.

Figure 5-167. EUR_CR_USE_FILTER2_EXTRA Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-168. EUR_CR_USE_FILTER2_EXTRA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	Reserved	R/W	000h	
19-4	CONSTANT	R/W	0000h	FIRH instruction filter option, unsigned integer additive constant
3-0	SHR	R/W	0h	FIRH instruction filter option, unsigned integer right shift amount

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.168 EUR_CR_PDS_PC_BASE Register (offset = B2Ch)

EUR_CR_PDS_PC_BASE is shown in Figure 5-168 and described in Table 5-169.

Figure 5-168. EUR_CR_PDS_PC_BASE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-169. EUR_CR_PDS_PC_BASE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	Reserved	R	0h	
29-0	ADDRESS	R	00000000h	PC base address

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.169 EUR_CR_USE_MOE Register (offset = B30h)

EUR_CR_USE_MOE is shown in Figure 5-169 and described in Table 5-170.

Figure 5-169. EUR_CR_USE_MOE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-170. EUR_CR_USE_MOE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R/W	000000h	
8	CFASEL	R/W	0h	Colour instruction format/address select
7-1	Reserved	R/W	00h	
0	FSFASEL	R/W	0h	EFO source format/address select

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.170 EUR_CR_EVENT_STATUS Register (offset = 12Ch)

There are 2 sources of external interrupts in the PowerVR SGX core: General system events and PDS cache invalidation events. The event status register indicates the source of any general event interrupt generated by PowerVR SGX. These events only result in an external interrupt if the relevant bit in the EUR_CR_EVENT_HOST_ENABLE register is set. Note1: This register must be combined with the EUR_CR_PDS_CACHE_STATUS register information to determine what has caused the external interrupt. Note2: The host can write to this register in which case any bits written as a '1' are ORed into the register i.e. the host can cause the core to generate an interrupt. (This is typically for debug purposes only)

EUR_CR_EVENT_STATUS is shown in Figure 5-170 and described in Table 5-171.

Figure 5-170. EUR_CR_EVENT_STATUS Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-171. EUR_CR_EVENT_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31	MASTER_INTERRUPT	R/W	0h	Status: PowerVR SGX Master Interrupt Control
30	Reserved	R/W	0h	
29	TIMER	R/W	0h	Status: The PDS Event timer interrupt (signalled on timer traversing through 0).
28	TA_DPM_FAULT	R/W	0h	Status: The TA Object Passing System has got out of sync, a Terminate Control Word DPM read was met by StartofBlock data
27	Reserved	R/W	0h	
26	MADD_CACHE_INVALCOMPLETE	R/W	0h	Status: Invalidate of Madd L0, L1, L2 Cache Complete
25	DPM_OUT_OF_MEMORY_ZLS	R/W	0h	Status: Memory Allocation Failed for ZLS
24	DPM_TA_MEM_FREE	R/W	0h	Status: All memory associated with TA context has been freed
23	ISP_END_TILE	R/W	0h	Status: ISP End tile
22	DPM_INITEND	R/W	0h	Status: DPM Initialisation Complete
21	OTPM_LOADED	R/W	0h	Status: The OTPM has loaded its state
20	OTPM_INV	R/W	0h	Status: The OTPM has invalidated its contents
19	OTPM_FLUSHED	R/W	0h	Status: The OTPM has flushed its state
18	PIXELBE_END_RENDER	R/W	0h	Status: End of Scene
17	ISP_HALT	R/W	0h	Status: ISP Halt breakpoint triggered
16	ISP_VISIBILITY_FAIL	R/W	0h	Status: Visibility Test Fail
15	BREAKPOINT	R/W	0h	Status: ISP Breakpoint
14	SW_EVENT	R/W	0h	Status: Software triggered event
13	TA_FINISHED	R/W	0h	Status: TA complete - This is set and held high after the TE terminates scene, until TA is restarted
12	TA_TERMINATE	R/W	0h	Status: The TA terminate complete (issue as part of complex scene) - This is set and held high after the TE aborts a MacroTile after failed DPM allocation request, until TA is restarted
11	TPC_CLEAR	R/W	0h	Status: The TPC has reset its state
10	TPC_FLUSH	R/W	0h	Status: The TPC has stored its state
9	DPM_CONTROL_CLEAR	R/W	0h	Status: The reset of the DPM control context has been completed
8	DPM_CONTROL_LOAD	R/W	0h	Status: The load of the DPM control context has been completed
7	DPM_CONTROL_STORE	R/W	0h	Status: The store of the DPM control context has been completed
6	DPM_STATE_CLEAR	R/W	0h	Status: The reset of the DPM state context has been completed
5	DPM_STATE_LOAD	R/W	0h	Status: The load of the DPM state context has been completed
4	DPM_STATE_STORE	R/W	0h	Status: The store of the DPM state context has been completed
3	DPM_REACHED_MEM_THRESH	R/W	0h	Status: DPM Memory allocation has reached the preset memory threshold for total memory allocation

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 5-171. EUR_CR_EVENT_STATUS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	DPM_OUT_OF_MEMORY_GBL	R/W	0h	Status: DPM Memory allocation failed for the global list
1	DPM_OUT_OF_MEMORY_MT	R/W	0h	Status: DPM Memory allocation failed for a macro tile
0	DPM_3D_MEM_FREE	R/W	0h	Status: DPM Memory allocation complete for the current render

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.171 EUR_CR_EVENT_HOST_ENABLE Register (offset = 130h)

This register enables interrupts. Writing a '1' to a bit field enables the relevant Event. All the bit fields correspond exactly to those in the EUR_CR_EVENT_STATUS register. The MASTER_INTERRUPT bit is a global enable which overrides the Event enables, i.e. '1' - enable external interrupts, '0' - disable external interrupts.

EUR_CR_EVENT_HOST_ENABLE is shown in Figure 5-171 and described in Table 5-172.

Figure 5-171. EUR_CR_EVENT_HOST_ENABLE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-172. EUR_CR_EVENT_HOST_ENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31	MASTER_INTERRUPT	R/W	0h	Host_Enable: PowerVR SGX Master Interrupt Control
30	Reserved	R/W	0h	
29	TIMER	R/W	0h	Host_Enable: The PDS Event timer interrupt (signalled on timer traversing through 0).
28	TA_DPM_FAULT	R/W	0h	Host_Enable: The Tiling Co-Processor Object Passing System has got out of sync, a Terminate Control Word DPM read was met by StartofBlock data
27	Reserved	R/W	0h	
26	MADD_CACHE_INVALCOMPLETE	R/W	0h	Host_Enable: Invalidate of Madd L0, L1, L2 Cache Complete
25	DPM_OUT_OF_MEMORY_ZLS	R/W	0h	Host_Enable: Memory Allocation Failed for ZLS
24	DPM_TA_MEM_FREE	R/W	0h	Host_Enable: All memory associated with TA context has been freed
23	ISP_END_TILE	R/W	0h	Host_Enable: ISP End tile
22	DPM_INITEND	R/W	0h	Host_Enable: DPM Initialisation Complete
21	OTPM_LOADED	R/W	0h	Host_Enable: The OTPM has loaded its state
20	OTPM_INV	R/W	0h	Host_Enable: The OTPM has invalidated its contents
19	OTPM_FLUSHED	R/W	0h	Host_Enable: The OTPM has flushed its state
18	PIXELBE_END_RENDER	R/W	0h	Host_Enable: End of Scene
17	ISP_HALT	R/W	0h	Host_Enable: ISP Halt breakpoint triggered
16	ISP_VISIBILITY_FAIL	R/W	0h	Host_Enable: Visibility Test Fail
15	BREAKPOINT	R/W	0h	Host_Enable: ISP Breakpoint
14	SW_EVENT	R/W	0h	Host_Enable: Software triggered event
13	TA_FINISHED	R/W	0h	Host_Enable: Tiling Co-Processor complete, (Scene Terminated)
12	TA_TERMINATE	R/W	0h	Host_Enable: The Tiling Co-Processor terminate complete (issue as part of complex scene) (MacroTile Aborted)
11	TPC_CLEAR	R/W	0h	Host_Enable: The TPC has reset its state
10	TPC_FLUSH	R/W	0h	Host_Enable: The TPC has stored its state
9	DPM_CONTROL_CLEAR	R/W	0h	Host_Enable: The reset of the DPM control context has been completed
8	DPM_CONTROL_LOAD	R/W	0h	Host_Enable: The load of the DPM control context has been completed
7	DPM_CONTROL_STORE	R/W	0h	Host_Enable: The store of the DPM control context has been completed
6	DPM_STATE_CLEAR	R/W	0h	Host_Enable: The reset of the DPM state context has been completed
5	DPM_STATE_LOAD	R/W	0h	Host_Enable: The load of the DPM state context has been completed
4	DPM_STATE_STORE	R/W	0h	Host_Enable: The store of the DPM state context has been completed
3	DPM_REACHED_MEM_THRESH	R/W	0h	Host_Enable: DPM Memory allocation has reached the preset memory threshold for total memory allocation

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 5-172. EUR_CR_EVENT_HOST_ENABLE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	DPM_OUT_OF_MEMORY_GBL	R/W	0h	Host_Enable: DPM Memory allocation failed for the global list
1	DPM_OUT_OF_MEMORY_MT	R/W	0h	Host_Enable: DPM Memory allocation failed for a macro tile
0	DPM_3D_MEM_FREE	R/W	0h	Host_Enable: DPM Memory allocation complete for the current render

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.172 EUR_CR_BIF_CTRL Register (offset = C00h)

EUR_CR_BIF_CTRL is shown in Figure 5-172 and described in Table 5-173.

Figure 5-172. EUR_CR_BIF_CTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-173. EUR_CR_BIF_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	Reserved	R/W	0000h	
15	MMU_BYPASS_USE	R/W	1h	Setting to '1' bypasses the MMU for this requestor
14	MMU_BYPASS_ISP	R/W	1h	Setting to '1' bypasses the MMU for this requestor
13	MMU_BYPASS_TSPP	R/W	1h	Setting to '1' bypasses the MMU for this requestor
12	MMU_BYPASS_PBE	R/W	1h	Setting to '1' bypasses the MMU for this requestor
11	Reserved	R/W	1h	
10	MMU_BYPASS_TE	R/W	1h	Setting to '1' bypasses the MMU for this requestor
9	MMU_BYPASS_VDM	R/W	1h	Setting to '1' bypasses the MMU for this requestor
8	MMU_BYPASS_CACHE	R/W	1h	Setting to '1' bypasses the MMU for this requestor
7-5	Reserved	R/W	0h	
4	CLEAR_FAULT	R/W	0h	Writing '1' clears the current fault in the EUR_CR_BIF_INT_STAT register
3	INVALDC	R/W	0h	Writing '1' causes flushing of directory cache (and all other caches and tlbs)
2	FLUSH	R/W	0h	Writing '1' causes flushing of address translation cache and all tlbs
1	PAUSE	R/W	0h	Writing '1' causes bif operation to pause
0	NOREORDER	R/W	0h	Writing '1' disables transaction re-ordering by signaling to the external SoC infrastructure to closely order the requests (Debug Only)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.173 EUR_CR_BIF_INT_STAT Register (offset = C04h)

EUR_CR_BIF_INT_STAT is shown in Figure 5-173 and described in Table 5-174.

Figure 5-173. EUR_CR_BIF_INT_STAT Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-174. EUR_CR_BIF_INT_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	Reserved	R	00000h	
14	PF_N_RW	R	0h	Indicates whether the current fault is a page or R/W protection fault
13-0	FAULT	R	0000h	1 bit per requestor indicating a page fault (only 1 can be set at once)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.174 EUR_CR_EVENT_HOST_CLEAR Register (offset = 134h)

This register is used to clear event interrupts. Writing a '1' to a bit field clears the relevant Event. All the bit fields correspond exactly to those in the EUR_CR_EVENT_STATUS register.

EUR_CR_EVENT_HOST_CLEAR is shown in Figure 5-174 and described in Table 5-175.

Figure 5-174. EUR_CR_EVENT_HOST_CLEAR Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-175. EUR_CR_EVENT_HOST_CLEAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31	MASTER_INTERRUPT	R/W	0h	Host_Clear: PowerVR SGX Master Interrupt
30	Reserved	R/W	0h	
29	TIMER	R/W	0h	Host_Clear: The PDS Event timer interrupt (signalled on timer traversing through 0).
28	TA_DPM_FAULT	R/W	0h	Host_Clear: The TA Object Passing System has got out of sync, a Terminate Control Word DPM read was met by StartofBlock data
27	Reserved	R/W	0h	
26	MADD_CACHE_INVALCOMPLETE	R/W	0h	Host_Clear: Invalidate of Madd L1, L2 Cache Complete
25	DPM_OUT_OF_MEMORY_ZLS	R/W	0h	Host_Clear: Memory Allocation Failed for ZLS
24	DPM_TA_MEM_FREE	R/W	0h	Host_Clear: All memory associated with TA context has been freed
23	ISP_END_TILE	R/W	0h	Host_Clear: ISP End tile
22	DPM_INITEND	R/W	0h	Host_Clear: DPM Initialisation Complete
21	OTPM_LOADED	R/W	0h	Host_Clear: The OTPM has loaded its state
20	OTPM_INV	R/W	0h	Host_Clear: The OTPM has invalidated its contents
19	OTPM_FLUSHED	R/W	0h	Host_Clear: The OTPM has flushed its state
18	PIXELBE_END_RENDER	R/W	0h	Host_Clear: End of Scene
17	ISP_HALT	R/W	0h	Host_Clear: ISP Halt breakpoint triggered
16	ISP_VISIBILITY_FAIL	R/W	0h	Host_Clear: Visibility Test Fail
15	BREAKPOINT	R/W	0h	Host_Clear: ISP Breakpoint
14	SW_EVENT	R/W	0h	Host_Clear: Software triggered event
13	TA_FINISHED	R/W	0h	Host_Clear: Tiling Co-Processor complete (Scene Terminated)
12	TA_TERMINATE	R/W	0h	Host_Clear: The Tiling Co-Processor terminate complete (issue as part of complex scene) (MacroTile aborted)
11	TPC_CLEAR	R/W	0h	Host_Clear: The TPC has reset its state
10	TPC_FLUSH	R/W	0h	Host_Clear: The TPC has stored its state
9	DPM_CONTROL_CLEAR	R/W	0h	Host_Clear: The reset of the DPM control context has been completed
8	DPM_CONTROL_LOAD	R/W	0h	Host_Clear: The load of the DPM control context has been completed
7	DPM_CONTROL_STORE	R/W	0h	Host_Clear: The store of the DPM control context has been completed
6	DPM_STATE_CLEAR	R/W	0h	Host_Clear: The reset of the DPM state context has been completed
5	DPM_STATE_LOAD	R/W	0h	Host_Clear: The load of the DPM state context has been completed
4	DPM_STATE_STORE	R/W	0h	Host_Clear: The store of the DPM state context has been completed
3	DPM_REACHED_MEM_THRESH	R/W	0h	Host_Clear: DPM Memory allocation has reached the preset memory threshold for total memory allocation
2	DPM_OUT_OF_MEMORY_GBL	R/W	0h	Host_Clear: DPM Memory allocation failed for the global list
1	DPM_OUT_OF_MEMORY_MT	R/W	0h	Host_Clear: DPM Memory allocation failed for a macro tile
0	DPM_3D_MEM_FREE	R/W	0h	Host_Clear: DPM Memory allocation complete for the current render

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.175 EUR_CR_BIF_FAULT Register (offset = C08h)

EUR_CR_BIF_FAULT is shown in Figure 5-175 and described in Table 5-176.

Figure 5-175. EUR_CR_BIF_FAULT Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-176. EUR_CR_BIF_FAULT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved	R	0h	
27-12	ADDR	R	0000h	Page-aligned address causing page fault
11-0	Reserved	R	000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.176 EUR_CR_PDS_CACHE_STATUS Register (offset = 138h)

There are 2 sources of external interrupts in the PowerVR SGX core: General system events and PDS cache invalidation events. The PDS cache status register indicates the source of any PDS cache invalidation event interrupts generated by PowerVR SGX. These events only result in an external interrupt if the relevant bit in the EUR_CR_PDS_CACHE_HOST_ENABLE register is set. Note1: This register must be combined with the EUR_CR_EVENT_STATUS register information to determine what has caused the external interrupt. Note2: The host can write to this register in which case any bits written as a '1' are ORed into the register i.e. the host can cause the core to generate an interrupt. (This is typically for debug purposes only)

EUR_CR_PDS_CACHE_STATUS is shown in Figure 5-176 and described in Table 5-177.

Figure 5-176. EUR_CR_PDS_CACHE_STATUS Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-177. EUR_CR_PDS_CACHE_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R/W	000000h	
8	DSC1_INV3	R/W	0h	Status: Context ID 3 has been invalidated in PDS Data Store Cache 1
7	DSC1_INV2	R/W	0h	Status: Context ID 2 has been invalidated in PDS Data Store Cache 1
6	DSC1_INV1	R/W	0h	Status: Context ID 1 has been invalidated in PDS Data Store Cache 1
5	DSC1_INV0	R/W	0h	Status: Context ID 0 has been invalidated in PDS Data Store Cache 1
4	DSC0_INV3	R/W	0h	Status: Context ID 3 has been invalidated in PDS Data Store Cache 0
3	DSC0_INV2	R/W	0h	Status: Context ID 2 has been invalidated in PDS Data Store Cache 0
2	DSC0_INV1	R/W	0h	Status: Context ID 1 has been invalidated in PDS Data Store Cache 0
1	DSC0_INV0	R/W	0h	Status: Context ID 0 has been invalidated in PDS Data Store Cache 0
0	CSC_INV	R/W	0h	Status: PDS Code Store Cache has been invalidated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.177 EUR_CR_PDS_CACHE_HOST_ENABLE Register (offset = 13Ch)

This register enables interrupts. Writing a '1' to a bit field enables the relevant Event. All the bit fields correspond exactly to those in the EUR_CR_PDS_CACHE_STATUS register. The MASTER_INTERRUPT bit contained in the EUR_CR_EVENT_HOST_ENABLE register is a global enable which overrides the Event enables, i.e. '1' - enable external interrupts, '0' - disable external interrupts.

EUR_CR_PDS_CACHE_HOST_ENABLE is shown in Figure 5-177 and described in Table 5-178.

Figure 5-177. EUR_CR_PDS_CACHE_HOST_ENABLE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-178. EUR_CR_PDS_CACHE_HOST_ENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R/W	000000h	
8	DSC1_INV3	R/W	0h	Host_Enable: Context ID 3 has been invalidated in PDS Data Store Cache 1
7	DSC1_INV2	R/W	0h	Host_Enable: Context ID 2 has been invalidated in PDS Data Store Cache 1
6	DSC1_INV1	R/W	0h	Host_Enable: Context ID 1 has been invalidated in PDS Data Store Cache 1
5	DSC1_INV0	R/W	0h	Host_Enable: Context ID 0 has been invalidated in PDS Data Store Cache 1
4	DSC0_INV3	R/W	0h	Host_Enable: Context ID 3 has been invalidated in PDS Data Store Cache 0
3	DSC0_INV2	R/W	0h	Host_Enable: Context ID 2 has been invalidated in PDS Data Store Cache 0
2	DSC0_INV1	R/W	0h	Host_Enable: Context ID 1 has been invalidated in PDS Data Store Cache 0
1	DSC0_INV0	R/W	0h	Host_Enable: Context ID 0 has been invalidated in PDS Data Store Cache 0
0	CSC_INV	R/W	0h	Host_Enable: PDS Code Store Cache has been invalidated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.178 EUR_CR_PDS_CACHE_HOST_CLEAR Register (offset = 140h)

This register is used to clear event interrupts. Writing a '1' to a bit field clears the relevant Event. All the bit fields correspond exactly to those in the EUR_CR_PDS_CACHE_STATUS register.

EUR_CR_PDS_CACHE_HOST_CLEAR is shown in Figure 5-178 and described in Table 5-179.

Figure 5-178. EUR_CR_PDS_CACHE_HOST_CLEAR Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-179. EUR_CR_PDS_CACHE_HOST_CLEAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R/W	000000h	
8	DSC1_INV3	R/W	0h	Host_Clear: Context ID 3 has been invalidated in PDS Data Store Cache 1
7	DSC1_INV2	R/W	0h	Host_Clear: Context ID 2 has been invalidated in PDS Data Store Cache 1
6	DSC1_INV1	R/W	0h	Host_Clear: Context ID 1 has been invalidated in PDS Data Store Cache 1
5	DSC1_INV0	R/W	0h	Host_Clear: Context ID 0 has been invalidated in PDS Data Store Cache 1
4	DSC0_INV3	R/W	0h	Host_Clear: Context ID 3 has been invalidated in PDS Data Store Cache 0
3	DSC0_INV2	R/W	0h	Host_Clear: Context ID 2 has been invalidated in PDS Data Store Cache 0
2	DSC0_INV1	R/W	0h	Host_Clear: Context ID 1 has been invalidated in PDS Data Store Cache 0
1	DSC0_INV0	R/W	0h	Host_Clear: Context ID 0 has been invalidated in PDS Data Store Cache 0
0	CSC_INV	R/W	0h	Host_Clear: PDS Code Store Cache has been invalidated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.179 EUR_CR_BIF_DIR_LIST_BASE0 Register (offset = C84h)

EUR_CR_BIF_DIR_LIST_BASE0 is shown in Figure 5-179 and described in Table 5-180.

Figure 5-179. EUR_CR_BIF_DIR_LIST_BASE0 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-180. EUR_CR_BIF_DIR_LIST_BASE0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	ADDR	R/W	00000h	Base address for Directory Cache updates
11-0	Reserved	R/W	000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.180 EUR_CR_BIF_TA_REQ_BASE Register (offset = C90h)

EUR_CR_BIF_TA_REQ_BASE is shown in Figure 5-180 and described in Table 5-181.

Figure 5-180. EUR_CR_BIF_TA_REQ_BASE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-181. EUR_CR_BIF_TA_REQ_BASE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved	R/W	0h	
27-20	ADDR	R/W	00h	Base address for ta requestor - 256MB, 1MB aligned
19-0	Reserved	R/W	00000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.181 EUR_CR_BIF_MEM_REQ_STAT Register (offset = CA8h)

EUR_CR_BIF_MEM_REQ_STAT is shown in Figure 5-181 and described in Table 5-182.

Figure 5-181. EUR_CR_BIF_MEM_REQ_STAT Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-182. EUR_CR_BIF_MEM_REQ_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	Reserved	R	000000h	
7-0	READS	R	00h	Outstanding read requests in the memif

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.182 EUR_CR_BIF_3D_REQ_BASE Register (offset = CACh)

EUR_CR_BIF_3D_REQ_BASE is shown in Figure 5-182 and described in Table 5-183.

Figure 5-182. EUR_CR_BIF_3D_REQ_BASE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-183. EUR_CR_BIF_3D_REQ_BASE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved	R/W	0h	
27-20	ADDR	R/W	00h	Base address for 3d requestors - 256MB, 1MB aligned
19-0	Reserved	R/W	00000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.183 EUR_CR_BIF_ZLS_REQ_BASE Register (offset = CB0h)

EUR_CR_BIF_ZLS_REQ_BASE is shown in Figure 5-183 and described in Table 5-184.

Figure 5-183. EUR_CR_BIF_ZLS_REQ_BASE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-184. EUR_CR_BIF_ZLS_REQ_BASE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved	R/W	0h	
27-20	ADDR	R/W	00h	Base address for zls requestor - 256MB, 1MB aligned
19-0	Reserved	R/W	00000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.184 EUR_CR_CLKGATESTATUS Register (offset = 4h)

Clock Gating Status reflects the condition of the clock gate controls for each module.

EUR_CR_CLKGATESTATUS is shown in Figure 5-184 and described in Table 5-185.

Figure 5-184. EUR_CR_CLKGATESTATUS Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-185. EUR_CR_CLKGATESTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	Reserved	R	000h	
20	USE_CLKS	R	0h	Status of the USSE clock 0 - USSE: USSE clock is gated and the module is inactive 1 - Clock: Clock to the USSE is running
19-17	Reserved	R	0h	
16	DPM_CLKS	R	0h	Status of the DPM clock 0 - DPM: DPM clock is gated and the module is inactive 1 - Clock: Clock to the DPM is running
15-13	Reserved	R	0h	
12	TA_CLKS	R	0h	Status of the TA clock 0 - TA: TA clock is gated and the module is inactive 1 - Clock: Clock to the TA is running
11-9	Reserved	R	0h	
8	TSP_CLKS	R	0h	Status of the TSP clock 0 - TSP: TSP clock is gated and the module is inactive 1 - Clock: Clock to the TSP is running
7-5	Reserved	R	0h	
4	ISP_CLKS	R	0h	Status of the ISP clock 0 - ISP: ISP clock is gated and the module is inactive 1 - Clock: Clock to the ISP is running
3-1	Reserved	R	0h	
0	Reserved	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.185 EUR_CR_VDM_START Register (offset = 200h)

Any write to this register starts the Vertex Data Master DMA operation

EUR_CR_VDM_START is shown in Figure 5-185 and described in Table 5-186.

Figure 5-185. EUR_CR_VDM_START Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-186. EUR_CR_VDM_START Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	W	00000000h	
0	PULSE	W	0h	Start VDM DMA

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.186 EUR_CR_TE_AA Register (offset = 204h)

This register controls the anti-aliasing mode of the Tiling Co-Processor, independent control is provided in both X and Y axis.

EUR_CR_TE_AA is shown in Figure 5-186 and described in Table 5-187.

Figure 5-186. EUR_CR_TE_AA Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-187. EUR_CR_TE_AA Register Field Descriptions

Bit	Field	Type	Reset	Description
31	X	R/W	0h	Anti-Aliasing in X: 0 - No: No Anti-Aliasing on X plane 1 - Yes: Anti-Aliasing on X plane
30	Y	R/W	0h	Anti-Aliasing in Y: 0 - No: No Anti-Aliasing on Y plane 1 - Yes: Anti-Aliasing on Y plane
29-0	Reserved	R/W	00000000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.187 EUR_CR_TE_MTILE1 Register (offset = 208h)

The Tiling Co-Processor allocates memory in macrotiles, which is a group of individual tiles. The number of macrotiles is selectable between 4 and 16, the boundaries of the macrotiles in the X axis are defined in this register.

EUR_CR_TE_MTILE1 is shown in Figure 5-187 and described in Table 5-188.

Figure 5-187. EUR_CR_TE_MTILE1 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-188. EUR_CR_TE_MTILE1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	NUMBER	R/W	0h	PSG_MacroTile_Number: 0 - 4: 4 MacroTiles 1 - 16: 16 MacroTiles
30	Reserved	R/W	0h	
29-22	X1	R/W	00h	X1 MacroTile boundary, left tile X for second column of macrotiles (16MT mode)
21-20	Reserved	R/W	0h	
19-12	X2	R/W	00h	X2 MacroTile boundary, left tile X for second(4MT)/third(16MT) column of macrotiles
11-8	Reserved	R/W	0h	
7-0	X3	R/W	00h	X3 MacroTile boundary, left tile X for fourth column of macrotiles (16MT)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.188 EUR_CR_TE_MTILE2 Register (offset = 20Ch)

The Tiling Co-Processor allocates memory in macrotiles, which is a group of individual tiles. The number of macrotiles is selectable between 4 and 16, the boundaries of the macrotiles in the Y axis are defined in this register.

EUR_CR_TE_MTILE2 is shown in Figure 5-188 and described in Table 5-189.

Figure 5-188. EUR_CR_TE_MTILE2 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-189. EUR_CR_TE_MTILE2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	Reserved	R/W	0h	
29-22	Y1	R/W	00h	Y1 MacroTile boundary, top tile Y for second row of macrotiles (16MT mode)
21-20	Reserved	R/W	0h	
19-12	Y2	R/W	00h	Y2 MacroTile boundary, top tile Y for second(4MT)/third(16MT) row of macrotiles
11-8	Reserved	R/W	0h	
7-0	Y3	R/W	00h	Y1 MacroTile boundary, top tile Y for fourth row of macrotiles (16MT mode)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.189 EUR_CR_TE_SCREEN Register (offset = 210h)

In order to perform the tiling operation and generate the display list the maximum screen size must be configured in terms of the number of tiles in X and Y axis.

EUR_CR_TE_SCREEN is shown in Figure 5-189 and described in Table 5-190.

Figure 5-189. EUR_CR_TE_SCREEN Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-190. EUR_CR_TE_SCREEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	Reserved	R/W	000h	
19-12	YMAX	R/W	00h	Maximum Y tile address visible on screen
11-8	Reserved	R/W	0h	
7-0	XMAX	R/W	00h	Maximum X tile address visible on screen

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.190 EUR_CR_TE_MTILE Register (offset = 214h)

This register defines the number of individual tiles within the macrotiles. This is used in the process of memory allocation.

EUR_CR_TE_MTILE is shown in Figure 5-190 and described in Table 5-191.

Figure 5-190. EUR_CR_TE_MTILE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-191. EUR_CR_TE_MTILE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R/W	0000h	
17-0	STRIDE	R/W	00000h	Number of tiles in a Macrotile. Stride = (XTile * YTiles)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.191 EUR_CR_TE_PSG Register (offset = 218h)

This register defines the global control for the Parameter Stream Generator within the Tiling Co-Processor. This module formats the display list generated by the Tiling Co-Processor.

EUR_CR_TE_PSG is shown in Figure 5-191 and described in Table 5-192.

Figure 5-191. EUR_CR_TE_PSG Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-192. EUR_CR_TE_PSG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	Reserved	R/W	00h	
24	ENABLE_CONTEXT_STATE_RESTORE	R/W	0h	Enables sampling of Driver TE_STATE_ISP_STATE_ID register on context switch/restore when set, when reset current local value is preserved.
23	ZONLYRENDER	R/W	0h	Don't invalidate Tail Pointer Cache entries on a Terminate command. Only effective when COMPLETEONTERMINATE is 0x0 0 - Do: Do Invalidate 1 - Don: Don't Invalidate
22	COMPLETEONTERMINATE	R/W	0h	0x1 Write region headers, terminate streams and invalidate tail pointer cache entries on terminate. 0 - If: If ZONLYRENDER = 0x0 then force an Interrupt, however if ZONLYRENDER = 0x1 then write region headers and terminate streams.
21	ZLOADENABLE	R/W	0h	If 0x1 set the zload enable bit in the region headers. If 0x0 the region header zload bit is controlled by the hardware automatically. The bit is set if a partial render has been performed on that tile.
20	ZSTOREENABLE	R/W	0h	If 0x1 set the zstore enable bit in the region headers. If 0x0 the region header zstore bit is controlled by the hardware automatically. The bit is set if a partial render is about to be performed on that tile.
19	PADZEROS	R/W	0h	Pad empty doublewords in allocation with zeros after Stream Link/Terminate. The allocation block size is 16 DWords, so potentially inserts 13 extra DWords of zeros into a pointer block. Debug only. Makes examination of memory more readable on a per frame basis.
18	FORCENEWSTATE	R/W	0h	Always embed state information in control stream. Debug only.
17-0	Reserved	R/W	00000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.192 EUR_CR_TE_PSGREGION_BASE Register (offset = 21Ch)

This register defines the base address in memory of the Region Header writes by the TA. Region headers are the first part of the display list and contain an entry per tile with information on global setup and a link address to parameters.

EUR_CR_TE_PSGREGION_BASE is shown in Figure 5-192 and described in Table 5-193.

Figure 5-192. EUR_CR_TE_PSGREGION_BASE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-193. EUR_CR_TE_PSGREGION_BASE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved	R/W	0h	
27-6	ADDR	R/W	000000h	256MB Addressable, 512-bit aligned Base Address for Region Header writes
5-0	Reserved	R/W	00h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.193 EUR_CR_TE_TPC_BASE Register (offset = 220h)

This register defines the base address in memory of the Tail Pointer Cache. A tail pointer is the current last address written to for a tiles individual display list, an entry for active tiles is maintained as primitives are processed by the TA.

EUR_CR_TE_TPC_BASE is shown in Figure 5-193 and described in Table 5-194.

Figure 5-193. EUR_CR_TE_TPC_BASE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-194. EUR_CR_TE_TPC_BASE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved	R/W	0h	
27-6	ADDR	R/W	000000h	256MB Addressable, 512-bit aligned Base Address for Tail Pointer Cache. The tail pointer is the current last address written to for a control stream for a tile.
5-0	Reserved	R/W	00h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.194 EUR_CR_TE_TPCCONTROL Register (offset = 224h)

The Tail Pointer Cache is used to keep track of the last address written to for a particular tile, this is stored in a mixture of on chip cache and external memory. Under certain circumstances it is necessary to either reset the cache or flush the on chip cache contents such that they are visible in memory.

EUR_CR_TE_TPCCONTROL is shown in Figure 5-194 and described in Table 5-195.

Figure 5-194. EUR_CR_TE_TPCCONTROL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-195. EUR_CR_TE_TPCCONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	CLEAR	W	0h	Reset contents of Tail Pointer Cache
30	FLUSH	W	0h	Flush contents of Tail Pointer Cache to Memory
29-0	Reserved	W	00000000h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.195 EUR_CR_TE_RGNBBOX_X Register (offset = 228h)

Reset or enabled by the MTE_RESETBBOX and MTE_UPDATEBBOX fields in the MTEControl word of the Input Parameter format. This is the maximum extent in X of the post transformed, clipped data before region clipping.

EUR_CR_TE_RGNBBOX_X is shown in Figure 5-195 and described in Table 5-196.

Figure 5-195. EUR_CR_TE_RGNBBOX_X Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-196. EUR_CR_TE_RGNBBOX_X Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	Reserved	R	00h	
24-16	MAX	R	000h	XMax value for maintained Region Generator Bounding Box
15-9	Reserved	R	00h	
8-0	MIN	R	1FFh	XMin value for maintained Region Generator Bounding Box

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.196 EUR_CR_TE_RGNBBOX_Y Register (offset = 22Ch)

Reset or enabled by the MTE_RESETBBOX and MTE_UPDATEBBOX fields in the MTE Control word of the Input Parameter format. This is the maximum extent in Y of the post transformed, clipped data before region clipping.

EUR_CR_TE_RGNBBOX_Y is shown in Figure 5-196 and described in Table 5-197.

Figure 5-196. EUR_CR_TE_RGNBBOX_Y Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-197. EUR_CR_TE_RGNBBOX_Y Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	Reserved	R	00h	
24-16	MAX	R	000h	YMax value for maintained Region Generator Bounding Box
15-9	Reserved	R	00h	
8-0	MIN	R	1FFh	YMin value for maintained Region Generator Bounding Box

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.197 EUR_CR_MTE_OTPM_CSM_FLUSH_BASE Register (offset = 230h)

The TA maintains internal state of the current information for each macro tile and global list regarding the vertex blocks written to memory. In certain circumstances during a context switch it is necessary to flush this information to external memory.

EUR_CR_MTE_OTPM_CSM_FLUSH_BASE is shown in Figure 5-197 and described in Table 5-198.

Figure 5-197. EUR_CR_MTE_OTPM_CSM_FLUSH_BASE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-198. EUR_CR_MTE_OTPM_CSM_FLUSH_BASE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved	R/W	0h	
27-4	ADDR	R/W	000000h	256MB Addressable, 512-bit aligned Base Address for Object Tail Pointer Memory Context Switchable Memory (OTPM CSM) Flush. This is the current point for each macro tile and global list, for the vertex blocks written to memory by the Tile Accelerator.
3-0	Reserved	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.198 EUR_CR_MTE_OTPM_CSM_LOAD_BASE Register (offset = 234h)

The TA maintains internal state of the current information for each macro tile and global list regarding the vertex blocks written to memory. In certain circumstances during a context switch it is necessary to load this information from external memory.

EUR_CR_MTE_OTPM_CSM_LOAD_BASE is shown in Figure 5-198 and described in Table 5-199.

Figure 5-198. EUR_CR_MTE_OTPM_CSM_LOAD_BASE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-199. EUR_CR_MTE_OTPM_CSM_LOAD_BASE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved	R/W	0h	
27-4	ADDR	R/W	000000h	256MB Addressable, 512-bit aligned Base Address for Object Tail Pointer Memory Context Switchable Memory (OTPM CSM) Load. This is the current point for each macro tile and global list, for the vertex blocks written to memory by the Tile Accelerator.
3-0	Reserved	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.199 EUR_CR_VDM_CTRL_STREAM_BASE Register (offset = 238h)

PowerVR SGX masters information from memory using advanced DMA, this register defines the base address of the Vertex Data input control stream in memory.

EUR_CR_VDM_CTRL_STREAM_BASE is shown in Figure 5-199 and described in Table 5-200.

Figure 5-199. EUR_CR_VDM_CTRL_STREAM_BASE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-200. EUR_CR_VDM_CTRL_STREAM_BASE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved	R/W	0h	
27-2	ADDR	R/W	0000000h	256MB Addressable, byte aligned Base Address for Vertex Data control streams
1-0	Reserved	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.200 EUR_CR_MTE_CTRL Register (offset = 23Ch)

This register controls the global setup of the Tiling Process.

EUR_CR_MTE_CTRL is shown in Figure 5-200 and described in Table 5-201.

Figure 5-200. EUR_CR_MTE_CTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-201. EUR_CR_MTE_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R/W	000000h	
8	NUM_PARTITIONS	R/W	0h	Maximum number of partitions which can be used per USSE pipeline: 0 - Two: Two 1 - Four: Four
7	OPENGL	R/W	0h	Select OpenGL or D3D mode 0 - D3D: D3D 1 - OpenGL: OpenGL
6	PSOCULL_DISABLE	R/W	0h	Disable for perfect small object culling 0 - en: Perfect small object culling enabled 1 - dis: Perfect small object culling disabled
5	WCLAMPEN	R/W	0h	Enable W clamping 0 - dis: W clamping disabled 1 - en: W clamping enabled
4-0	GLOBALMACROTILETHRESH	R/W	00h	When writing to global list, number of primitives which are within the same macrotile before the macrotile calculation will change from global.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.201 EUR_CR_MTE_WCOMPARE Register (offset = 240h)

EUR_CR_MTE_WCOMPARE is shown in Figure 5-201 and described in Table 5-202.

Figure 5-201. EUR_CR_MTE_WCOMPARE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-202. EUR_CR_MTE_WCOMPARE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VALUE	R/W	00000000h	Compare value for W clamping. See the Input parameter format, viewport transform for details. Compare is applied post viewport transform.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.202 EUR_CR_MTE_WCLAMP Register (offset = 244h)

EUR_CR_MTE_WCLAMP is shown in Figure 5-202 and described in Table 5-203.

Figure 5-202. EUR_CR_MTE_WCLAMP Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-203. EUR_CR_MTE_WCLAMP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	VALUE	R/W	00000000h	Clamp value for W clamping. See the Input parameter format, viewport transform for details. Clamp is applied post viewport transform, if the w value is less than the WCOMPARE value.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.203 EUR_CR_MTE_SCREEN Register (offset = 248h)

In order to perform the tiling operation and generate the display list the maximum screen size must be configured in terms of the number of pixels in X and Y axis since this may not be the same as the number of tiles defined in the EUR_CR_TE_SCREEN register.

EUR_CR_MTE_SCREEN is shown in Figure 5-203 and described in Table 5-204.

Figure 5-203. EUR_CR_MTE_SCREEN Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-204. EUR_CR_MTE_SCREEN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	Reserved	R/W	00h	
23-12	PIXYMAX	R/W	000h	Screen height in pixels.
11-0	PIXXMAX	R/W	000h	Screen width in pixels.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.204 EUR_CR_MTE_OTPM_OP Register (offset = 24Ch)

The TA maintains internal state of the current information for each macro tile and global list regarding the vertex blocks written to memory. In certain circumstances during a context switch it is necessary to load or flush this information to external memory.

EUR_CR_MTE_OTPM_OP is shown in Figure 5-204 and described in Table 5-205.

Figure 5-204. EUR_CR_MTE_OTPM_OP Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-205. EUR_CR_MTE_OTPM_OP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	Reserved	W	00000000h	
2	CSM_FLUSH	W	0h	Flush CSM contents to CSM Flush Address
1	CSM_LOAD	W	0h	Load CSM contents from CSM Load Address
0	CSM_INV	W	0h	Invalidate CSM contents

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.205 EUR_CR_MTE_MULTISAMPLECTL Register (offset = 250h)

This applies when anti-aliasing is enabled. This is used to calculate whether an anti-aliased triangle can be perfect small object culled.

EUR_CR_MTE_MULTISAMPLECTL is shown in Figure 5-205 and described in Table 5-206.

Figure 5-205. EUR_CR_MTE_MULTISAMPLECTL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-206. EUR_CR_MTE_MULTISAMPLECTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	MSAA_Y3	R/W	0h	Unsigned sub-pixel offset for the 4th multisample Y position
27-24	MSAA_X3	R/W	0h	Unsigned sub-pixel offset for the 4th multisample X position
23-20	MSAA_Y2	R/W	0h	Unsigned sub-pixel offset for the 3rd multisample Y position
19-16	MSAA_X2	R/W	0h	Unsigned sub-pixel offset for the 3rd multisample X position
15-12	MSAA_Y1	R/W	0h	Unsigned sub-pixel offset for the 2nd multisample Y position
11-8	MSAA_X1	R/W	0h	Unsigned sub-pixel offset for the 2nd multisample X position
7-4	MSAA_Y0	R/W	0h	Unsigned sub-pixel offset for the 1st multisample Y position
3-0	MSAA_X0	R/W	0h	Unsigned sub-pixel offset for the 1st multisample X position

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.206 EUR_CR_MTE_SIG1 Register (offset = 26Ch)

EUR_CR_MTE_SIG1 is shown in Figure 5-206 and described in Table 5-207.

Figure 5-206. EUR_CR_MTE_SIG1 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-207. EUR_CR_MTE_SIG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TE_SIGNATURE	R	00000000h	Signature of output to TE. A signature is an accumulative LFSR (Linear Feedback Shift Register) of the valid pipeline data using an XOR structure.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.207 EUR_CR_MTE_SIG2 Register (offset = 270h)

EUR_CR_MTE_SIG2 is shown in Figure 5-207 and described in Table 5-208.

Figure 5-207. EUR_CR_MTE_SIG2 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-208. EUR_CR_MTE_SIG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	MEM_SIGNATURE	R	00000000h	Signature of output to memory. A signature is an accumulative LFSR (Linear Feedback Shift Register) of the valid pipeline data using an XOR structure.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.208 EUR_CR_TE_STATE Register (offset = 274h)

This register is read on VDM_START_PULSE and should be saved and updated by the driver app by app EUR_CR_TE_STATE is shown in Figure 5-208 and described in Table 5-209.

Figure 5-208. EUR_CR_TE_STATE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-209. EUR_CR_TE_STATE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	Reserved	R/W	000h	
20-16	ISP_STATE_ID	R/W	0h	This register contains the PSG internal ISP State ID Counter value. It must be saved off by the driver on TA terminate or finished and restored before vdm_start_pulse.
15-0	MTILE	R/W	0000h	A read of this register returns the page allocated to the host, when valid. If it is not valid the return data is undefined.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.209 EUR_CR_PERF Register (offset = 40h)

Within SGX there are a number of performance counters which enable profiling of an application in terms of the low level hardware performance; this register enables selection of specific groups of performance counters and the ability to reset those counters. Full details can be found in the performance profiling section of this document. Write a '1' followed by a '0' to the _CLR fields to reset the specific counter

EUR_CR_PERF is shown in Figure 5-209 and described in Table 5-210.

Figure 5-209. EUR_CR_PERF Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-210. EUR_CR_PERF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	Reserved	R/W	00000h	
12-9	COUNTER_SELECT	R/W	Fh	0x0 Core Overview counter group. Default value is set to 0xF, no counter will be in operation 1 - Primitive: Primitive Clipping counter group 2 - Scene: Scene Buffer counter group 3 - ISP: ISP counter group 4 - TSP: TSP counter group 5 - USE: USE counter group 6 - Texturing: Texturing counter group 7 - Reserved_7: Reserved 8 - Reserved_8: Reserved 9 - Core: Core Overview 2 counter group
8	COUNTER_8_CLR	R/W	0h	Reset counter = '1', normal counter operation '0'
7	COUNTER_7_CLR	R/W	0h	Reset counter = '1', normal counter operation '0'
6	COUNTER_6_CLR	R/W	0h	Reset counter = '1', normal counter operation '0'
5	COUNTER_5_CLR	R/W	0h	Reset counter = '1', normal counter operation '0'
4	COUNTER_4_CLR	R/W	0h	Reset counter = '1', normal counter operation '0'
3	COUNTER_3_CLR	R/W	0h	Reset counter = '1', normal counter operation '0'
2	COUNTER_2_CLR	R/W	0h	Reset counter = '1', normal counter operation '0'
1	COUNTER_1_CLR	R/W	0h	Reset counter = '1', normal counter operation '0'
0	COUNTER_0_CLR	R/W	0h	Reset counter = '1', normal counter operation '0'

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.210 EUR_CR_PERF_COUNTER0 Register (offset = 44h)

Performance counter 0 status register

EUR_CR_PERF_COUNTER0 is shown in Figure 5-210 and described in Table 5-211.

Figure 5-210. EUR_CR_PERF_COUNTER0 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-211. EUR_CR_PERF_COUNTER0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COUNTER0	R	00000000h	32 bit integer count value, see performance profiling section for details

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.211 EUR_CR_PERF_COUNTER1 Register (offset = 48h)

Performance counter 1 status register

EUR_CR_PERF_COUNTER1 is shown in Figure 5-211 and described in Table 5-212.

Figure 5-211. EUR_CR_PERF_COUNTER1 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-212. EUR_CR_PERF_COUNTER1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COUNTER1	R	00000000h	32 bit integer count value, see performance profiling section for details

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.212 EUR_CR_PERF_COUNTER2 Register (offset = 4Ch)

Performance counter 2 status register

EUR_CR_PERF_COUNTER2 is shown in Figure 5-212 and described in Table 5-213.

Figure 5-212. EUR_CR_PERF_COUNTER2 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-213. EUR_CR_PERF_COUNTER2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COUNTER2	R	00000000h	32 bit integer count value, see performance profiling section for details

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.213 EUR_CR_CLIP_SIG1 Register (offset = 304h)

EUR_CR_CLIP_SIG1 is shown in Figure 5-213 and described in Table 5-214.

Figure 5-213. EUR_CR_CLIP_SIG1 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-214. EUR_CR_CLIP_SIG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CLIP_SIGNATURE	R	00000000h	Signature of Barycentric vertices output during each clipping pass i.e. clipped against each clip plane in turn. A signature is an accumulative LFSR (Linear Feedback Shift Register) of the valid pipeline data using an XOR structure.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.214 EUR_CR_PIXELBE_SIG01 Register (offset = 308h)

EUR_CR_PIXELBE_SIG01 is shown in Figure 5-214 and described in Table 5-215.

Figure 5-214. EUR_CR_PIXELBE_SIG01 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-215. EUR_CR_PIXELBE_SIG01 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SIGNATURE	R	00000000h	Signature of data output from PBE to BIF. A signature is an accumulative LFSR (Linear Feedback Shift Register) of the valid pipeline data using an XOR structure.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.215 EUR_CR_PIXELBE_SIG02 Register (offset = 30Ch)

EUR_CR_PIXELBE_SIG02 is shown in Figure 5-215 and described in Table 5-216.

Figure 5-215. EUR_CR_PIXELBE_SIG02 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-216. EUR_CR_PIXELBE_SIG02 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	SIGNATURE	R	00000000h	Signature of control data output from PBE to BIF. A signature is an accumulative LFSR (Linear Feedback Shift Register) of the valid pipeline data using an XOR structure.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.216 EUR_CR_CLKGATECTLOVR Register (offset = 8h)

Core module clock gating override controls: allows clocks to be enabled temporarily for register writes by hosts.

EUR_CR_CLKGATECTLOVR is shown in Figure 5-216 and described in Table 5-217.

Figure 5-216. EUR_CR_CLKGATECTLOVR Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-217. EUR_CR_CLKGATECTLOVR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	Reserved	R/W	000h	
21-20	USE_CLKO	R/W	0h	Clock gating control override for the USSE 0 - Reserved_0: Reserved 1 - Enabled: Host enable clock to the USSE request 2 - Reserved_2: Reserved 3 - Disabled: Host disable clock to USSE request. USSE clock may be gated if not other outstanding host requests.
19-18	Reserved	R/W	0h	
17-16	DPM_CLKO	R/W	0h	Clock gating control override for the DPM 0 - Reserved_0: Reserved 1 - Enabled: Clock to the DPM is enabled 2 - Reserved_2: Reserved 3 - Disabled: Host disable clock to DPM request. DPM clock may be gated if not other outstanding host requests.
15-14	Reserved	R/W	0h	
13-12	TA_CLKO	R/W	0h	Clock gating control override for the TA 0 - Reserved_0: Reserved 1 - Enabled: Clock to the TA is enabled 2 - Reserved_2: Reserved 3 - Disabled: Host disable clock to TA request. TA clock may be gated if not other outstanding host requests.
11-10	Reserved	R/W	0h	
9-8	TSP_CLKO	R/W	0h	Clock gating control override for the TSP 0 - Reserved_0: Reserved 1 - Enabled: Clock to the TSP is enabled 2 - Reserved_2: Reserved 3 - Disabled: Host disable clock to TSP request. TSP clock may be gated if no other outstanding host requests.
7-6	Reserved	R/W	0h	
5-4	ISP_CLKO	R/W	0h	Clock gating control override for the ISP 0 - Reserved_0: Reserved 1 - Enabled: Clock to the ISP is enabled 2 - Reserved_2: Reserved 3 - Disabled: Host disable clock to ISP request. ISP clock may be gated if no other outstanding host requests.
3-2	Reserved	R/W	0h	
1-0	Reserved	R/W	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.217 EUR_CR_PERF_COUNTER3 Register (offset = 50h)

Performance counter 3 status register

EUR_CR_PERF_COUNTER3 is shown in Figure 5-217 and described in Table 5-218.

Figure 5-217. EUR_CR_PERF_COUNTER3 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-218. EUR_CR_PERF_COUNTER3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COUNTER3	R	00000000h	32 bit integer count value, see performance profiling section for details

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.218 EUR_CR_VDM_CONTEXT_STORE_START Register (offset = 330h)

Start VDM Context store, pulse kicks off the store operation. Status flags when this is complete and the stream and index position registers are valid

EUR_CR_VDM_CONTEXT_STORE_START is shown in Figure 5-218 and described in Table 5-219.

Figure 5-218. EUR_CR_VDM_CONTEXT_STORE_START Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-219. EUR_CR_VDM_CONTEXT_STORE_START Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R/W	00000000h	
0	PULSE	R/W	0h	Start VDM Context store

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.219 EUR_CR_VDM_CONTEXT_STORE_STREAM Register (offset = 334h)

Stores the stream position when the context store occured

EUR_CR_VDM_CONTEXT_STORE_STREAM is shown in Figure 5-219 and described in Table 5-220.

Figure 5-219. EUR_CR_VDM_CONTEXT_STORE_STREAM Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-220. EUR_CR_VDM_CONTEXT_STORE_STREAM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved	R	0h	
27-2	ADDR	R	0000000h	stream position for context store, number of 32bit dwords from a base address update (ie. between stream link or terminates)
1-0	Reserved	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.220 EUR_CR_VDM_CONTEXT_STORE_INDEX Register (offset = 338h)

Stores the index position when the context store occured

EUR_CR_VDM_CONTEXT_STORE_INDEX is shown in Figure 5-220 and described in Table 5-221.

Figure 5-220. EUR_CR_VDM_CONTEXT_STORE_INDEX Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-221. EUR_CR_VDM_CONTEXT_STORE_INDEX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	Reserved	R	000h	
21-0	ADDR	R	000000h	Index position for context store, number of index words from an index define

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.221 EUR_CR_VDM_CONTEXT_STORE_STATUS Register (offset = 33Ch)

Signals when a context store is complete and the stream and index location registers are valid

EUR_CR_VDM_CONTEXT_STORE_STATUS is shown in Figure 5-221 and described in Table 5-222.

Figure 5-221. EUR_CR_VDM_CONTEXT_STORE_STATUS Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-222. EUR_CR_VDM_CONTEXT_STORE_STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R	000000h	
8	NA	R	0h	Context store not applicable doing a stream terminate
7-5	Reserved	R	0h	
4	PROCESS	R	0h	Context store processing
3-1	Reserved	R	0h	
0	COMPLETE	R	0h	Context store complete

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.222 EUR_CR_VDM_CONTEXT_STORE_STATE0 Register (offset = 340h)

Terminate state control registers used for creating the terminate PDS task.

EUR_CR_VDM_CONTEXT_STORE_STATE0 is shown in Figure 5-222 and described in Table 5-223.

Figure 5-222. EUR_CR_VDM_CONTEXT_STORE_STATE0 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-223. EUR_CR_VDM_CONTEXT_STORE_STATE0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	Reserved	R/W	0h	
29	TAPDSSTATE_TERMINATE	R/W	0h	defines the last word in the control stream, used to generate end ta
28	TAPDSSTATE_COMPLEX	R/W	0h	defines if the state update is complex
27-22	Reserved	R/W	00h	
21-0	TAPDSSTATE_BASEADDR	R/W	000000h	Base address (128Bit Aligned, 256 MByte range), in 16 byte units of the PDS code and data to be executed

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.223 EUR_CR_VDM_CONTEXT_STORE_STATE1 Register (offset = 344h)

Terminate state control registers used for creating the terminate PDS task.

EUR_CR_VDM_CONTEXT_STORE_STATE1 is shown in Figure 5-223 and described in Table 5-224.

Figure 5-223. EUR_CR_VDM_CONTEXT_STORE_STATE1 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-224. EUR_CR_VDM_CONTEXT_STORE_STATE1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	TAPDSSTATE_DATASIZE	R/W	00h	Size of the PDS Data segment in 128 bit words. The data segment always precedes the code segment. 0 = 0x0, Max = 0x20
25	Reserved	R/W	0h	
24	TAPDSSTATE_DEBUG	R/W	0h	Enables breakpoints on this program
23-18	Reserved	R/W	00h	
17	TAPDSSTATE_MTE_EMIT	R/W	0h	If set the USSE will output state to the MTE
16	TAPDSSTATE_SEC_EXEC	R/W	0h	0 ' Do not execute secondary task on the USSE/1 ' Execute secondary task on the USSE Has no effect on primary tasks.
15	TAPDSSTATE_SD	R/W	0h	Indicates a sequential dependency
14-13	TAPDSSTATE_USEPIPE	R/W	0h	0 ' Reserved/1 ' Run on pipe 1/2 ' Run on pipe 2/3 ' Run on both pipes
12	TAPDSSTATE_PARTIAL	R/W	0h	Indicates partial secondary updates.
11-9	TAPDSSTATE_PARTITIONS	R/W	0h	Number of output partitions required. 0 = 0x0 and Max = 0x4
8	TAPDSSTATE_SECONDARY	R/W	0h	Indicates that the allocation is secondary
7-0	TAPDSSTATE_USEATTRIBUTESIZE	R/W	00h	Size of the USSE Attribute Requirements for the task in 128 bit Words

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.224 EUR_CR_PERF_COUNTER4 Register (offset = 54h)

Performance counter 4 status register

EUR_CR_PERF_COUNTER4 is shown in Figure 5-224 and described in Table 5-225.

Figure 5-224. EUR_CR_PERF_COUNTER4 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-225. EUR_CR_PERF_COUNTER4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COUNTER4	R	00000000h	32 bit integer count value, see performance profiling section for details

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.225 EUR_CR_VDM_WAIT_FOR_KICK Register (offset = 348h)

Indicates the status on the control stream fetch

EUR_CR_VDM_WAIT_FOR_KICK is shown in Figure 5-225 and described in Table 5-226.

Figure 5-225. EUR_CR_VDM_WAIT_FOR_KICK Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-226. EUR_CR_VDM_WAIT_FOR_KICK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R	00000000h	
0	STATUS	R	1h	Indicates when state machine is in idle

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.226 EUR_CR_VDM_CONTEXT_STORE_WRAP_INDEX Register (offset = 34Ch)

Stores the number of indices since the last wrap

EUR_CR_VDM_CONTEXT_STORE_WRAP_INDEX is shown in Figure 5-226 and described in Table 5-227.

Figure 5-226. EUR_CR_VDM_CONTEXT_STORE_WRAP_INDEX Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-227. EUR_CR_VDM_CONTEXT_STORE_WRAP_INDEX Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	Reserved	R	0h	
21-0	COUNT	R	0h	Wrap Index Count for context store, number of indices since last wrap

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.227 EUR_CR_VDM_CONTEXT_STORE_WRAPPED Register (offset = 350h)

Indicates the number of times that index wrapping has occurred after a context store

EUR_CR_VDM_CONTEXT_STORE_WRAPPED is shown in Figure 5-227 and described in Table 5-228.

Figure 5-227. EUR_CR_VDM_CONTEXT_STORE_WRAPPED Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-228. EUR_CR_VDM_CONTEXT_STORE_WRAPPED Register Field Descriptions

Bit	Field	Type	Reset	Description
31-22	Reserved	R	0h	
21-0	COUNT	R	0h	Wrapped count for context store, number of times the hardware has wrapped after a context store

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.228 EUR_CR_VDM_CONTEXT_STORE_FAN Register (offset = 354h)

The driver sets this register on a context switch such that the hardware can resume knowing where the centre point of a fan is located or the winding order of the strip, expressed as an index.

EUR_CR_VDM_CONTEXT_STORE_FAN is shown in Figure 5-228 and described in Table 5-229.

Figure 5-228. EUR_CR_VDM_CONTEXT_STORE_FAN Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-229. EUR_CR_VDM_CONTEXT_STORE_FAN Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	R/W	0h	
25	INDEX3	R/W	0h	Controls setting of the winding order for the primitive about to be resumed
24	INDEX2	R/W	0h	An enable bit, 1 = valid and triggers the use of the index indicated in bits 23:0. (Valid only on a VDM_START_PULSE)
23-0	INDEX1	R/W	0h	Context store, index to be used as center point to Fan after a context store.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.229 EUR_CR_VDM_CONTEXT_STORE_WRAP Register (offset = 358h)

Indicates the number of times that index wrapping has occurred after a context store

EUR_CR_VDM_CONTEXT_STORE_WRAP is shown in Figure 5-229 and described in Table 5-230.

Figure 5-229. EUR_CR_VDM_CONTEXT_STORE_WRAP Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 5-230. EUR_CR_VDM_CONTEXT_STORE_WRAP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	Reserved	R/W	0h	
23	RESUME3	R/W	0h	Indicates control of the value to be used in the next index block (1 = context wrap counter + 1, 0 = reset context wrap counter)
22	RESUME2	R/W	0h	An enable bit, 1 = valid and triggers the use of the wrap count value indicated in bits 21:0. (Valid only on a VDM_START_PULSE)
21-0	RESUME1	R/W	0h	Context store, wrap counter value used on a resume

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.230 EUR_CR_PERF_COUNTER5 Register (offset = 58h)

Performance counter 5 status register

EUR_CR_PERF_COUNTER5 is shown in Figure 5-230 and described in Table 5-231.

Figure 5-230. EUR_CR_PERF_COUNTER5 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-231. EUR_CR_PERF_COUNTER5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COUNTER5	R	00000000h	32 bit integer count value, see performance profiling section for details

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.231 EUR_CR_PERF_COUNTER6 Register (offset = 5Ch)

Performance counter 6 status register

EUR_CR_PERF_COUNTER6 is shown in Figure 5-231 and described in Table 5-232.

Figure 5-231. EUR_CR_PERF_COUNTER6 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-232. EUR_CR_PERF_COUNTER6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COUNTER6	R	00000000h	32 bit integer count value, see performance profiling section for details

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5.5.1.232 EUR_CR_PERF_COUNTER7 Register (offset = 60h)

Performance counter 7 status register

EUR_CR_PERF_COUNTER7 is shown in Figure 5-232 and described in Table 5-233.

Figure 5-232. EUR_CR_PERF_COUNTER7 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 5-233. EUR_CR_PERF_COUNTER7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COUNTER7	R	00000000h	32 bit integer count value, see performance profiling section for details

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Chapter 6

SPRUH73–July 2011

Memory Subsystem

This chapter describes the memory subsystem of the device.

Topic **Page**

6.1 GPMC **848**
6.2 OCMC-RAM **1000**
6.3 EMIF **1003**
6.4 ELM **1059**

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1 GPMC

6.1.1 Introduction

The general-purpose memory controller (GPMC) is an unified memory controller dedicated to interfacing external memory devices:

- Asynchronous SRAM-like memories and application-specific integrated circuit (ASIC) devices
- Asynchronous, synchronous, and page mode (only available in non-multiplexed mode) burst NOR flash devices
- NAND Flash
- Pseudo-SRAM devices

6.1.1.1 GPMC Features

The general features of the GPMC module include:

- Data path to external memory device can be 16- or 8-bit wide
- 32-bit OCPIP 2.0 compliant core, single slave interface. Support non-wrapping and wrapping burst up to 16x32bits.
- Support secure transactions controlled by security firewall
- Up to 62.5 MHz external memory clock performance (single device) with L3 clock of 125 MHz
- Support for the following memory types:
  - External asynchronous or synchronous 8-bit width memory or device (non burst device)
  - External asynchronous or synchronous 16-bit width memory or device
  - External 16-bit non-multiplexed NOR Flash device
  - External 16-bit address and data multiplexed NOR Flash device
  - External 8-bit and 16-bit NAND flash device
  - External 16-bit pSRAM device
- Support of Error Code detection using BCH code (t=4, 8 or 16) or Hamming code for 8-bit or 16-bit NAND-flash, organized with page size of 512 Byte, 1Kbytes or more.
- Support 1 GBytes maximum addressing capability which can be divided into 8 independent chip-select with programmable bank size and base address on 16 MBytes, 32 MBytes, 64 MBytes, 128 MBytes boundary
- Fully pipelined operation for optimal memory bandwidth usage
- Support external device clock frequency of 1, 2, 3 and 4 divider from L3 clock.
- Support programmable auto-clock gating when there is no access.
- Support Midlereq/SidleAck protocol
- Support the following interface protocols when communicating with external memory or external devices.
  - Asynchronous read/write access
  - Asynchronous read page access (4-8-16 Word16)
  - Synchronous read/write access
  - Synchronous read burst access without wrap capability (4-8-16 Word16)
  - Synchronous read burst access with wrap capability (4-8-16 Word16)
- Address and Data multiplexed access
- Each chip-select as independent and programmable control signal timing parameters for Setup and Hold time. Parameters are set according to the memory device timing parameters, with one L3 clock cycle timing granularity.
- Flexible internal access time control (wait state) and flexible handshake mode using external WAIT pins monitoring (up to 4 WAIT pins)
- Support bus keeping

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

- Support bus turn around
- Pre-fetch and write posting engine associated with system DMA to get full performance from NAND device with minimum impact on NOR/SRAM concurrent access.
- On the fly ECC Hamming Code calculation to improve NAND usage reliability with minimum impact on SW

6.1.1.2 Block Diagram

The GPMC can access various external devices through the L3 Slow Interconnect. The flexible programming model allows a wide range of attached device types and access schemes. Based on the programmed configuration bit fields stored in the GPMC registers, the GPMC is able to generate all control signals timing depending on the attached device and access type. Given the chip-select decoding and its associated configuration registers, the GPMC selects the appropriate device type control signals timing.

Figure 6-1 shows the GPMC functional block diagram. The GPMC consists of six blocks:

- Interconnect port interface
- Address decoder, GPMC configuration, and chip-select configuration register file
- Access engine
- Prefetch and write-posting engine
- Error correction code engine (ECC)
- External device/memory port interface

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 6-1. GPMC Block Diagram

6.1.1.3 Unsupported GPMC Features

The following module features are not supported in this device.

Table 6-1. Unsupported GPMC Features

Feature	Reason
Chip Select 7	Not pinned out
32-bit devices	Only 16 data lines pinned out
WAIT[3:2]	Not pinned out. All CS regions must use WAIT0 or WAIT1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.2 Integration

An instantiation of GPMC provides this device with access to NAND Flash, NOR Flash, and other asynchronous and synchronous interface peripherals. Figure 6-2 shows the integration of the GPMC module in this device.

Figure 6-2. GPMC Integration

6.1.2.1 GPMC Connectivity Attributes

The general connectivity attributes for the GPMC module are shown in Table 6-2.

Table 6-2. GPMC Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_L3S_GCLK
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	1 interrupt to Sabertooth
DMA Requests	1 DMA request to EDMA
Physical Address	L3 Slow Slave Port Memory and control register regions qualified with MAddressSpace bit

6.1.2.2 GPMC Clock and Reset Management

The GPMC is a synchronous design and operates from the same clock as the Slow L3. All timings use this clock as a reference.

Table 6-3. GPMC Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
prcm_gpmc_clk Interface / Functional clock	100 MHz	CORE_COLKOUTM4 / 2	pd_per_l3s_gclk From PRCM

6.1.2.3 GPMC Pin List

The GPMC external interface signals are shown in Table 6-4.

Table 6-4. GPMC Pin List

Pin	Type	Description
GPMC_A[27:0]	O	Address outputs
GPMC_AD[15:0]	I/O	Data - multiplexed with Address[16:1] and Address[27:12]
GPMC_CS[7:0]n	O	Chip selects (active low)
GPMC_CLK	O	Synchronous mode clock
GPMC_CLKRET	I	Synchronous mode return clock
GPMC_ADVn_ALE	O	Address Valid or Address Latch Enable depending if NOR or NAND protocol memories are selected.
GPMC_OEn_REn	O	Output Enable (active low). Also used as Read Enable (active low) for NAND protocol memories
GPMC_WEn	O	Write Enable (active low)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 6-4. GPMC Pin List (continued)

Pin	Type	Description
GPMC_BE0n_CLE	O	Lower Byte Enable (active low). Also used as Command Latch Enable for NAND protocol memories
GPMC_BE1n	O	Upper Byte Enable (active low)
GPMC_WPn	O	Write Protect (active low)
GPMC_WAIT[3:0]	I	External wait signal for NOR and NAND protocol memories.
GPMC_IO_DIR2	O	GPMC.D[15:0] signal direction control (high when transmit – for external xcvr)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3 Functional Description

6.1.3.1 GPMC Signals

Table 6-5 lists the GPMC subsystem I/O pins.

NOTE: The $\overline{\text{GPMC_CS[7:6]}}$ and GPMC_WAIT1 pins are not available on all devices, refer to your device-specific data manual for availability.

Table 6-5. GPMC I/O Description

Pin Name	I/O	Description
GPMC_A[27:0]	O	Address
GPMC_D[15:0]	I/O	Data
$\overline{\text{GPMC_CS[7:0]}}$	O	Chip-selects (active low)
GPMC_CLK	O	Clock generated for the external memory or device
GPMC_$\overline{\text{ADV}}$_ALE	O	Address valid (active low). Also used as address latch enable (active high) for NAND protocol memories.
$\overline{\text{GPMC_OE_RE}}$	O	Output enable (active low). Also used as read enable (active low) for NAND protocol memories.
$\overline{\text{GPMC_WE}}$	O	Write enable (active low)
GPMC_$\overline{\text{BE0}}$_CLE	O	Lower-byte enable (active low). Also used as command latch enable for NAND protocol memories.
$\overline{\text{GPMC_BE1}}$	O	Upper-byte enable (active low).
$\overline{\text{GPMC_WP}}$	O	Write protect (active low)
GPMC_WAIT[1:0]	I	External wait signal for NOR and NAND protocol memories. The wait signals can be mapped on any of the chip-select.
GPMC_DIR	O	GPMC_D[15:0] signal direction control: Low during transmit (for write access: data OUT from GPMC to memory) High during receive (for read access: data IN from memory to GPMC)

Table 6-6 shows the use of address and data GPMC controller pins based on the type of external device.

Table 6-6. GPMC Pin Multiplexing Options

GPMC Pin	Non Multiplexed Address Data 16-Bit Device	Non Multiplexed Address Data 8-Bit Device	Multiplexed Address Data 16-Bit Device	16-Bit NAND Device	8-Bit NAND Device
GPMC_A[27]	A26	A27	A26	Not Used	Not Used
GPMC_A[26]	A25	A26	A25	Not Used	Not Used
GPMC_A[25]	A24	A25	A24	Not Used	Not Used
GPMC_A[24]	A23	A24	A23	Not Used	Not Used
GPMC_A[23]	A22	A23	A22	Not Used	Not Used
GPMC_A[22]	A21	A22	A21	Not Used	Not Used
GPMC_A[21]	A20	A21	A20	Not Used	Not Used
GPMC_A[20]	A19	A20	A19	Not Used	Not Used
GPMC_A[19]	A18	A19	A18	Not Used	Not Used
GPMC_A[18]	A17	A18	A17	Not Used	Not Used
GPMC_A[17]	A16	A17	A16	Not Used	Not Used
GPMC_A[16]	A15	A16	Not Used	Not Used	Not Used
GPMC_A[15]	A14	A15	Not Used	Not Used	Not Used
GPMC_A[14]	A13	A14	Not Used	Not Used	Not Used
GPMC_A[13]	A12	A13	Not Used	Not Used	Not Used

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 6-6. GPMC Pin Multiplexing Options (continued)

GPMC Pin	Non Multiplexed Address Data 16-Bit Device	Non Multiplexed Address Data 8-Bit Device	Multiplexed Address Data 16-Bit Device	16-Bit NAND Device	8-Bit NAND Device
GPMC_A[12]	A11	A12	Not Used	Not Used	Not Used
GPMC_A[11]	A10	A11	Not Used	Not Used	Not Used
GPMC_A[10]	A9	A10	Not Used	Not Used	Not Used
GPMC_A[9]	A8	A9	Not Used	Not Used	Not Used
GPMC_A[8]	A7	A8	Not Used	Not Used	Not Used
GPMC_A[7]	A6	A7	Not Used	Not Used	Not Used
GPMC_A[6]	A5	A6	Not Used	Not Used	Not Used
GPMC_A[5]	A4	A5	Not Used	Not Used	Not Used
GPMC_A[4]	A3	A4	Not Used	Not Used	Not Used
GPMC_A[3]	A2	A3	Not Used	Not Used	Not Used
GPMC_A[2]	A1	A2	Not Used	Not Used	Not Used
GPMC_A[1]	A0	A1	Not Used	Not Used	Not Used
GPMC_A[0]	Not Used	A0	Not Used	Not Used	Not Used
GPMC_D[15]	D15	Not Used	A16/D15	D15	Not Used
GPMC_D[14]	D14	Not Used	A15/D14	D14	Not Used
GPMC_D[13]	D13	Not Used	A14/D13	D13	Not Used
GPMC_D[12]	D12	Not Used	A13/D12	D12	Not Used
GPMC_D[11]	D11	Not Used	A12/D11	D11	Not Used
GPMC_D[10]	D10	Not Used	A11/D10	D10	Not Used
GPMC_D[9]	D9	Not Used	A10/D9	D9	Not Used
GPMC_D[8]	D8	Not Used	A9/D8	D8	Not Used
GPMC_D[7]	D7	D7	A8/D7	D7	D7
GPMC_D[6]	D6	D6	A7/D6	D6	D6
GPMC_D[5]	D5	D5	A6/D5	D5	D5
GPMC_D[4]	D4	D4	A5/D4	D4	D4
GPMC_D[3]	D3	D3	A4/D3	D3	D3
GPMC_D[2]	D2	D2	A3/D2	D2	D2
GPMC_D[1]	D1	D1	A2/D1	D1	D1
GPMC_D[0]	D0	D0	A1/D0	D0	D0

With all device types, the GPMC does not drive unnecessary address lines. They stay at their reset value of 00.

Address mapping supports address/data-multiplexed 16-bit wide devices:

- The NOR flash memory controller still supports non-multiplexed address and data memory devices.
- Multiplexing mode can be selected through the GPMC_CONFIG1_i[9-8] MUXADDDATA bit field (i = 0 to 7).
- Asynchronous page mode is not supported for multiplexed address and data devices.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.2 GPMC Modes

This section shows three GPMC external connections options:

- Figure 6-3 shows a connection between the GPMC and a 16-bit synchronous address/data-multiplexed (or AAD-multiplexed, but this protocol use less address pins) external memory device.
- Figure 6-4 shows a connection between the GPMC and a 16-bit synchronous nonmultiplexed external memory device .
- Figure 6-5 shows a connection between the GPMC and a 8-bit NAND device

Figure 6-3. GPMC to 16-Bit Address/Data-Multiplexed Memory

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 6-4. GPMC to 16-Bit Nonmultiplexed Memory

Figure 6-5. GPMC to 8-Bit NAND Device

Copyright © 2011, Texas Instruments Incorporated

6.1.3.3 GPMC Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

- No STANDBY hardware handshake
- No wake-up request
- One system direct memory access (eDMA) request
- One interrupt request to the Cortex-A8 MPU Interrupt Controller (MA_IRQ)
- One clock for functional and interface domains

Figure 6-6 shows GPMC integration. Table 6-7 through Table 6-9 summarize the integration of the module in the device.

Figure 6-6. GPMC Integration

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 6-7. GPMC Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-up Capability	Interconnect
GPMC	ALWAYS_ON	No	L3_Slow

Table 6-8. GPMC Clocks and Resets

Module Instance	Destination Signal Name	Source Signal Name	Source	Description
Clocks				
GPMC	GPMC_FCLK	prcm_GPMC_CLK	PRCM	Functional clock
Resets				
GPMC	GPMC_RST	ALW_DOM_RST	PRCM	GPMC reset

Table 6-9. GPMC Hardware Requests

Module Instance	Source Signal Name	Destination Signal Name	Destination	Description
Interrupt Requests				
GPMC	GPMC_IRQ	A_IRQ_100	Cortex-A8	GPMC interrupt to Cortex-A8 MPU subsystem
DMA Requests				
GPMC	GPMC_DMA_REQ	e_DMA_52	eDMA	GPMC request from Prefetch Engine to eDMA

6.1.3.4 GPMC Functional Description

The GPMC basic programming model offers maximum flexibility to support various access protocols for each of the eight configurable chip-selects. Use optimal chip-select settings, based on the characteristics of the external device:

- Different protocols can be selected to support generic asynchronous or synchronous random-access devices (NOR flash, SRAM) or to support specific NAND devices.
- The address and the data bus can be multiplexed on the same external bus.
- Read and write access can be independently defined as asynchronous or synchronous.
- System requests (byte, 16-bit word, burst) are performed through single or multiple accesses. External access profiles (single, multiple with optimized burst length, native- or emulated-wrap) are based on external device characteristics (supported protocol, bus width, data buffer size, native-wrap support).
- System burst read or write requests are synchronous-burst (multiple-read or multiple-write). When neither burst nor page mode is supported by external memory or ASIC devices, system burst read or write requests are translated to successive single synchronous or asynchronous accesses (single reads or single writes). 8-bit wide devices are supported only in single synchronous or single asynchronous read or write mode.
- To simulate a programmable internal-wait state, an external wait pin can be monitored to dynamically control external access at the beginning (initial access time) of and during a burst access.

Each control signal is controlled independently for each chip-select. The internal functional clock of the GPMC (GPMC_FCLK) is used as a time reference to specify the following:

- Read- and write-access duration
- Most GPMC external interface control-signal assertion and deassertion times
- Data-capture time during read access
- External wait-pin monitoring time

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

- Duration of idle time between accesses, when required

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.1 GPMC Clock Configuration

Table 6-10 describes the GPMC clocks.

Table 6-10. GPMC Clocks

Signal	I/O	Description
GPMC_FCLK	I	Functional and interface clock
GPMC_CLK	O	External clock provided to synchronous external memory devices.

The GPMC_CLK is generated by the GPMC from the internal GPMC_FCLK clock. The source of the GPMC_FCLK is described in Table 6-8. The GPMC_CLK is configured via the GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER field (for i = 0 to 3) as shown in Table 6-11.

Table 6-11. GPMC_CONFIG1_i Configuration

Source Clock	GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER	GPMC_CLK Generated Clock Provided to External Memory Device
GPMC_FCLK	00	GPMC_FCLK
	01	GPMC_FCLK/2
	10	GPMC_FCLK/4
	11	GPMC_FCLK/8

6.1.3.4.2 GPMC Software Reset

The GPMC can be reset by software through the GPMC_SYSCONFIG[1] SOFTRESET bit. Setting the bit to 1 enables an active software reset that is functionally equivalent to a hardware reset. Hardware and software resets initialize all GPMC registers and the finite state-machine (FSM) immediately and unconditionally. The GPMC_SYSSTATUS[0] RESETDONE bit indicates that the software reset is complete when its value is 1. The software must ensure that the software reset completes before doing GPMC operations.

6.1.3.4.3 GPMC Power Management

GPMC power is supplied by the CORE power domain, and GPMC power management complies with system power-management guidelines. Table 6-12 describes power-management features available for the GPMC module.

Table 6-12. GPMC Local Power Management Features

Feature	Registers	Description
Clock Auto Gating	GPMC_SYSCONFIG[0] AUTOIDLE] bit	This bit allows a local power optimization inside the module, by gating the GPMC_FCLK clock upon the internal activity.
Slave Idle Modes	GPMC_SYSCONFIG[4-3] SIDLEMODE bit field	Force-idle, No-idle and Smart-idle wakeup modes are available
Clock Activity	N/A	Feature not available
Master Standby Modes	N/A	Feature not available
Global Wake-up Enable	N/A	Feature not available
Wake-up Sources Enable	N/A	Feature not available

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.4 GPMC Interrupt Requests

The GPMC generates one interrupt event as shown in Figure 6-6.

- The interrupt request goes from GPMC (GPMC_IRQ) to the Cortex-A8 MPU subsystem: A_IRQ_100

Table 6-13 lists the event flags, and their mask, that can cause module interrupts.

Table 6-13. GPMC Interrupt Events

Event Flag	Event Mask	Sensitivity	Map to	Description
GPMC_IRQSTATUS[9] WAIT1EDGEDETECTIONSTATUS	GPMC_IRQENABLE[9] WAIT1EDGEDETECTIONENABLE	Edge	A_IRQ_100	Wait1 edge detection interrupt: Triggered if a rising or falling edge is detected on the GPMC_WAIT1 signal. The rising or falling edge detection of Wait1 is selected through GPMC_CONFIG[9] WAIT1PINPOLARITY bit.
GPMC_IRQSTATUS[8] WAIT0EDGEDETECTIONSTATUS	GPMC_IRQENABLE[8] WAIT0EDGEDETECTIONENABLE	Edge	A_IRQ_100	Wait0 edge detection interrupt: Triggered if a rising or falling edge is detected on the GPMC_WAIT0 signal. The rising or falling edge detection of Wait0 is selected through GPMC_CONFIG[8] WAIT0PINPOLARITY bit.
GPMC_IRQSTATUS[1] TERMINALCOUNTSTATUS	GPMC_IRQENABLE[1] TERMINALCOUNTENABLE	Level	A_IRQ_100	Terminal count event: Triggered on prefetch process completion, that is when the number of currently remaining data to be requested reaches 0.
GPMC_IRQSTATUS[0] FIFOEVENTSTATUS	GPMC_IRQENABLE[0] FIFOEVENTENABLE	Level	A_IRQ_100	FIFO event interrupt: Indicates FIFO levels availability for in Write-Posting mode and prefetch mode. GPMC_PREFETCH_CONFIG[2] DMAMODE bit shall be cleared to 0.

6.1.3.4.5 GPMC DMA Requests

The GPMC generates one DMA event as shown in Figure 6-6.

- From GPMC (GPMC_DMA_REQ) to the eDMA: e_DMA_53

6.1.3.4.6 L3 Slow Interconnect Interface

The GPMC L3 Slow interconnect interface is a pipelined interface including an 16 × 32-bit word write buffer. Any system host can issue external access requests through the GPMC. The device system can issue the following requests through this interface:

- One 8-bit / 16-bit / 32-bit interconnect access (read/write)
- Two incrementing 32-bit interconnect accesses (read/write)
- Two wrapped 32-bit interconnect accesses (read/write)
- Four incrementing 32-bit interconnect accesses (read/write)
- Four wrapped 32-bit interconnect accesses (read/write)
- Eight incrementing 32-bit interconnect accesses (read/write)
- Eight wrapped 32-bit interconnect accesses (read/write)

Only linear burst transactions are supported; interleaved burst transactions are not supported. Only power-of-two-length precise bursts 2 × 32, 4 × 32, 8 × 32 or 16 × 32 with the burst base address aligned on the total burst size are supported (this limitation applies to incrementing bursts only).

This interface also provides one interrupt and one DMA request line, for specific event control.

It is recommended to program the GPMC_CONFIG1_i ATTACHEDDEVICEPAGELENGTH field ([24-23]) according to the effective attached device page length and to enable the GPMC_CONFIG1_i WRAPBURST bit ([31]) if the attached device supports wrapping burst. However, it is possible to emulate wrapping burst on a non-wrapping memory by providing relevant addresses within the page or splitting transactions. Bursts larger than the memory page length are chopped into multiple bursts transactions. Due to the alignment requirements, a page boundary is never crossed.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.7 GPMC Address and Data Bus

The current application supports GPMC connection to NAND devices and to address/data-multiplexed memories or devices. Connection to address/data-nonmultiplexed memories Depending on the GPMC configuration of each chip-select, address and data bus lines that are not required for a particular access protocol are not updated (changed from current value) and are not sampled when input (input data bus).

- For address/data-multiplexed and AAD-multiplexed NOR devices, the address is multiplexed on the data bus.
- 8-bit wide NOR devices do not use GPMC I/O: GPMC_D[15-8] for data (they are used for address if needed).
- 16-bit wide NAND devices do not use GPMC I/O: GPMC_A[27-0].
- 8-bit wide NAND devices do not use GPMC I/O: GPMC_A[27-0] and GPMC I/O: GPMC_D[15-8].

6.1.3.4.7.1 GPMC I/O Configuration Setting

NOTE: In this section and next sections, the i in GPMC_CONFIGx_i stands for the GPMC chip-select i where i = 0 to 7.

To select a NAND device, program the following register fields:

- GPMC_CONFIG1_i[11-10] DEVICETYPE field = 10b
- GPMC_CONFIG1_i[9-8] MUXADDDATA bit = 00

To select an address/data-multiplexed device, program the following register fields:

- GPMC_CONFIG1_i[11-10] DEVICETYPE field = 00
- GPMC_CONFIG1_i[9-8] MUXADDDATA bit = 10b

To select an address/address/data-multiplexed device, program the following register fields:

- GPMC_CONFIG1_i[11-10] DEVICETYPE field = 00
- GPMC_CONFIG1_i[9-8] MUXADDDATA bit = 01b

To select an address/data-nonmultiplexed device , program the following register fields:

- GPMC_CONFIG1_i[11-10] DEVICETYPE field = 00
- GPMC_CONFIG1_i[9-8] MUXADDDATA bit = 00

6.1.3.4.8 Address Decoder and Chip-Select Configuration

Addresses are decoded accordingly with the address request of the chip-select and the content of the chip-select base address register file, which includes a set of global GPMC configuration registers and eight sets of chip-select configuration registers.

The GPMC configuration register file is memory-mapped and can be read or written with byte, 16-bit word, or 32-bit word accesses. The register file should be configured as a noncacheable, nonbufferable region to prevent any desynchronization between host execution (write request) and the completion of register configuration (write completed with register updated). Section 6.1.5 provides the GPMC register locations. For the map of GPMC memory locations, see Table 6-59.

After the chip-select is configured, the access engine accesses the external device, drives the external interface control signals, and applies the interface protocol based on user-defined timing parameters and settings.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.8.1 Chip-Select Base Address and Region Size

Any external memory or ASIC device attached to the GPMC external interface can be accessed by any device system host within the GPMC 512-Mbyte contiguous address space. For details, see Table 6-59.

The GPMC 512 Mbyte address space can be divided into a maximum of eight chip-select regions with programmable base address and programmable CS size. The CS size is programmable from 16 Mbytes to 256 Mbytes (must be a power-of-2) and is defined by the mask field. Attached memory smaller than the programmed CS region size is accessed through the entire CS region (aliasing).

Each chip-select has a 6-bit base address encoding and a 4-bit decoding mask, which must be programmed according to the following rules:

- The programmed chip-select region base address must be aligned on the chip-select region size address boundary and is limited to a power-of-2 address value. During access decoding, the register base address value is used for address comparison with the address-bit line mapping as described in Figure 6-7 (with A0 as the device system byte-address line). Base address is programmed through the GPMC_CONFIG7_i[5-0] BASEADDRESS bit field.
- The register mask is used to exclude some address lines from the decoding. A register mask bit field cleared to 0 suppresses the associated address line from the address comparison (incoming address bit line is don't care). The register mask value must be limited to the subsequent value, based on the desired chip-select region size. Any other value has an undefined result. When multiple chip-select regions with overlapping addresses are enabled concurrently, access to these chip-select regions is cancelled and a GPMC access error is posted. The mask field is programmed through the GPMC_CONFIG7_i[11-8] MASKADDRESS bit field.

Figure 6-7. Chip-Select Address Mapping and Decoding Mask

Chip-select configuration (base and mask address or any protocol and timing settings) must be performed while the associated chip-select is disabled through the GPMC_CONFIG7_i[6] CSVALID bit (where i stands for the GPMC chip-select value, i = 0 to 7). In addition, a chip-select configuration can only be disabled if there is no ongoing access to that chip-select. This requires activity monitoring of the prefetch or write-posting engine if the engine is active on the chip-select. Also, the write buffer state must be monitored to wait for any posted write completion to the chip-select.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Any access attempted to a nonvalid GPMC address region (CSVALID disabled or address decoding outside a valid chip-select region) is not propagated to the external interface and a GPMC access error is posted. In case of chip-selects overlapping, an error is generated and no access will occur on either chip-select. Chip-select 0 is the only chip-select region enabled after either a power-up or a GPMC reset.

Although the GPMC interface can drive up to 8 chip-selects, the frequency specified for this interface is for a specific load. If this load is exceeded, the maximum frequency cannot be reached. One solution is to implement a board with buffers, to allow the slowest device to maintain the total load on the lines.

6.1.3.4.8.2 Access Protocol

6.1.3.4.8.2.1 Supported Devices

The access protocol of each chip-select can be independently specified through the GPMC_CONFIG1_i[11-10] DEVICETYPE parameter (where i = 0 to 7) for:

- Random-access synchronous or asynchronous memory like NOR flash, SRAM
- NAND flash asynchronous devices

For more information about the NAND flash GPMC basic programming model and NAND support, see Section 6.1.3.4.12 and Section 6.1.3.4.12.1.

6.1.3.4.8.2.2 Access Size Adaptation and Device Width

Each chip-select can be independently configured through the GPMC_CONFIG1_i[13-12] DEVICESIZE field (i = 0 to 7) to interface with a 16-bit wide device or an 8-bit wide device. System requests with data width greater than the external device data bus width are split into successive accesses according to both the external device data-bus width and little-endian data organization.

The device does not provide the A0 byte address line required for random-byte addressable 8-bit wide device interfacing (for both multiplexed and nonmultiplexed protocol). It limits the use of 8-bit wide device interfacing to byte-alias accesses. This limitation is not applicable to NAND device interfacing (8-bit wide or 16-bit wide devices).

6.1.3.4.8.2.3 Address/Data-Multiplexing Interface

For random synchronous or asynchronous memory interfacing (DEVICETYPE = 0b00), an address- and data-multiplexing protocol can be selected through the GPMC_CONFIG1_i[[9-8] MUXADDDATA bit field (i = 0 to 7). The $\overline{ADV}$ signal must be used as the external device address latch control signal. For the associated chip-select configuration, $\overline{ADV}$ assertion and deassertion time and $\overline{OE}$ assertion time must be set to the appropriate value to meet the address latch setup/hold time requirements of the external device (see Section 6.1.3.3).

This address/data-multiplexing interface is not applicable to NAND device interfacing. NAND devices require a specific address, command, and data multiplexing protocol (see Section 6.1.3.4.12).

6.1.3.4.8.3 External Signals

6.1.3.4.8.3.1 WAIT Pin Monitoring Control

GPMC access time can be dynamically controlled using an external gpmc_wait pin when the external device access time is not deterministic and cannot be defined and controlled only using the GPMC internal RDACCESSTIME, WRACCESSTIME and PAGEBURSTACCESSTIME wait state generator.

The GPMC features two input wait pin:gpmc_wait1, and gpmc_wait0. This pin allow control of external devices with different wait-pin polarity. They also allow the overlap of wait-pin assertion from different devices without affecting access to devices for which the wait pin is not asserted.

- The GPMC_CONFIG1_i[17-16] WAITPINSELECT bit field (where i = 0 to 7) selects which input gpmc_wait pin is used for the device attached to the corresponding chip-select.
- The polarity of the wait pin is defined through the WAITxPINPOLARITY bit of the GPMC_CONFIG register. A wait pin configured to be active low means that low level on the WAIT signal indicates that the data is not ready and that the data bus is invalid. When WAIT is inactive, data is valid.

Copyright © 2011, Texas Instruments Incorporated

The GPMC access engine can be configured per CS to monitor the wait pin of the external memory device or not, based on the access type: read or write.

- The GPMC_CONFIG1_i[22] WAITREADMONITORING bit defines whether the wait pin should be monitored during read accesses or not.
- The GPMC_CONFIG1_i[21] WAITWRITEMONITORING bit defines whether the wait pin should be monitored during write accesses or not.

The GPMC access engine can be configured to monitor the wait pin of the external memory device asynchronously or synchronously with the GPMC_CLK clock, depending on the access type: synchronous or asynchronous (the GPMC_CONFIG1_i[29] READTYPE and GPMC_CONFIG1_i[27] WRITETYPE bits).

6.1.3.4.8.3.2 Wait Monitoring During an Asynchronous Read Access

When wait-pin monitoring is enabled for read accesses (WAITREADMONITORING), the effective access time is a logical AND combination of the RDACCESSTIME timing completion and the wait-deasserted state.

During asynchronous read accesses with wait-pin monitoring enabled, the wait pin must be at a valid level (asserted or deasserted) for at least two GPMC clock cycles before RDACCESSTIME completes, to ensure correct dynamic access-time control through wait-pin monitoring. The advance pipelining of the two GPMC clock cycles is the result of the internal synchronization requirements for the WAIT signal.

In this context, RDACCESSTIME is used as a WAIT invalid timing window and is set to such a value that the wait pin is at a valid state two GPMC clock cycles before RDACCESSTIME completes.

Similarly, during a multiple-access cycle (for example, asynchronous read page mode), the effective access time is a logical AND combination of PAGEBURSTACCESSTIME timing completion and the wait-deasserted state. Wait-monitoring pipelining is also applicable to multiple accesses (access within a page).

- WAIT monitored as active freezes the CYCLETIME counter. For an access within a page, when the CYCLETIME counter is by definition in a lock state, WAIT monitored as asserted extends the current access time in the page. Control signals are kept in their current state. The data bus is considered invalid, and no data are captured during this clock cycle.
- WAIT monitored as inactive unfreezes the CYCLETIME counter. For an access within a page, when the CYCLETIME counter is by definition in a lock state, WAIT monitored as inactive completes the current access time and starts the next access phase in the page. The data bus is considered valid, and data are captured during this clock cycle. In case of a single access or if this was the last access in a multiple-access cycle, all signals are controlled according to their related control timing value and according to the CYCLETIME counter status.

When a delay larger than two GPMC clocks must be observed between wait-pin deactivation time and data valid time (including the required GPMC and the device data setup time), an extra delay can be added between wait-pin deassertion time detection and effective data-capture time and the effective unlock of the CYCLETIME counter. This extra delay can be programmed in the GPMC_CONFIG1_i[19-18] WAITMONITORINGTIME field (i = 0 to 7).

- The WAITMONITORINGTIME parameter does not delay the wait-pin active or inactive detection, nor does it modify the two GPMC clocks pipelined detection delay.
- This extra delay is expressed as a number of GPMC_CLK clock cycles, even though the access is defined as asynchronous, and no GPMC_CLK clock is provided to the external device. Still, GPMCFCLKDIVIDER is used as a divider for the GPMC clock, so it must be programmed to define the correct WAITMONITORINGTIME delay.

Figure 6-8 shows wait behavior during an asynchronous single read access.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 6-8. Wait Behavior During an Asynchronous Single Read Access (GPMCFCLKDivider = 1)

The WAIT signal is active low. GPMC_CONFIG1_i[19-18] WAITMONITORINGTIME = 00b or 01b.

6.1.3.4.8.3.3 Wait Monitoring During an Asynchronous Write Access

When wait-pin monitoring is enabled for write accesses (GPMC_CONFIG1_i[21] WAITWRITEMONITORING bit = 1), the WAIT-invalid timing window is defined by the WRACCESSTIME field. WRACCESSTIME must be set so that the wait pin is at a valid state two GPMC clock cycles before WRACCESSTIME completes. The advance pipelining of the two GPMC clock cycles is the result of the internal synchronization requirements for the WAIT signal.

- WAIT monitored as active freezes the CYCLETIME counter. This informs the GPMC that the data bus is not captured by the external device. The control signals are kept in their current state. The data bus still drives the data.
- WAIT monitored as inactive unfreezes the CYCLETIME counter. This informs that the data bus is correctly captured by the external device. All signals, including the data bus, are controlled according to their related control timing value and to the CYCLETIME counter status.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

When a delay larger than two GPMC clock cycles must be observed between wait-pin deassertion time and the effective data write into the external device (including the required GPMC data setup time and the device data setup time), an extra delay can be added between wait-pin deassertion time detection and effective data write time into the external device and the effective unfreezing of the CYCLETIME counter. This extra delay can be programmed in the GPMC_CONFIG1_i[19-18] WAITMONITORINGTIME fields (i = 0 to 7).

- The WAITMONITORINGTIME parameter does not delay the wait-pin assertion or deassertion detection, nor does it modify the two GPMC clock cycles pipelined detection delay.
- This extra delay is expressed as a number of GPMC_CLK clock cycles, even though the access is defined as asynchronous, and even though no clock is provided to the external device. Still, GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER is used as a divider for the GPMC clock and so it must be programmed to define the correct WAITMONITORINGTIME delay.

6.1.3.4.8.3.4 Wait Monitoring During a Synchronous Read Access

During synchronous accesses with wait-pin monitoring enabled, the wait pin is captured synchronously with GPMC_CLK, using the rising edge of this clock.

The WAIT signal can be programmed to apply to the same clock cycle it is captured in. Alternatively, it can be sampled one or two GPMC_CLK cycles ahead of the clock cycle it applies to. This pipelining is applicable to the entire burst access, and to all data phase in the burst access. This WAIT pipelining depth is programmed in the GPMC_CONFIG1_i[19-18] WAITMONITORINGTIME field (where i = 0 to 7), and is expressed as a number of GPMC_CLK clock cycles.

In synchronous mode, when wait-pin monitoring is enabled (GPMC_CONFIG1_i[22] WAITREADMONITORING bit), the effective access time is a logical AND combination of the RDACCESSTIME timing completion and the WAIT deasserted-state detection.

Depending on the programmed WAITMONITORINGTIME value, the wait pin should be at a valid level, either asserted or deasserted:

- In the same clock cycle the data is valid if WAITMONITORINGTIME = 0 (at RDACCESSTIME completion)
- In the WAITMONITORINGTIME x (GPMCFCLKDIVIDER + 1) GPMC_FCLK clock cycles before RDACCESSTIME completion if WAITMONITORINGTIME not equal to 0

Similarly, during a multiple-access cycle (burst mode), the effective access time is a logical AND combination of PAGEBURSTACCESSTIME timing completion and the wait-inactive state. The Wait pipelining depth programming applies to the whole burst access.

- WAIT monitored as active freezes the CYCLETIME counter. For an access within a burst (when the CYCLETIME counter is by definition in a lock state), WAIT monitored as active extends the current access time in the burst. Control signals are kept in their current state. The data bus is considered invalid, and no data are captured during this clock cycle.
- WAIT monitored as inactive unfreezes the CYCLETIME counter. For an access within a burst (when the CYCLETIME counter is by definition in lock state), WAIT monitored as inactive completes the current access time and starts the next access phase in the burst. The data bus is considered valid, and data are captured during this clock cycle. In a single access or if this was the last access in a multiple-access cycle, all signals are controlled according to their relative control timing value and the CYCLETIME counter status.

Figure 6-9 shows wait behavior during a synchronous read burst access.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 6-9. Wait Behavior During a Synchronous Read Burst Access

The WAIT signal is active low. WAITMONITORINGTIME = 00b or 01b.

6.1.3.4.8.3.5 Wait Monitoring During a Synchronous Write Access

During synchronous accesses with wait-pin monitoring enabled (the WAITWRITEMONITORING bit), the wait pin is captured synchronously with GPMC_CLK, using the rising edge of this clock.

If enabled, external wait-pin monitoring can be used in combination with WRACCESSTIME to delay the effective memory device GPMC_CLK capture edge.

Wait-monitoring pipelining depth is similar to synchronous read access:

- At WRACCESSTIME completion if WAITMONITORINGTIME = 0
- In the WAITMONITORINGTIME x (GPMCFCLKDIVIDER + 1) GPMC_FCLK cycles before WRACCESSTIME completion if WAITMONITORINGTIME not equal to 0.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Wait-monitoring pipelining definition applies to whole burst accesses:

- WAIT monitored as active freezes the CYCLETIME counter. For accesses within a burst, when the CYCLETIME counter is by definition in a lock state, WAIT monitored as active indicates that the data bus is not being captured by the external device. Control signals are kept in their current state. The data bus is kept in its current state.
- WAIT monitored as inactive unfreezes the CYCLETIME counter. For accesses within a burst, when the CYCLETIME counter is by definition in a lock state, WAIT monitored as inactive indicates the effective data capture of the bus by the external device and starts the next access of the burst. In case of a single access or if this was the last access in a multiple access cycle, all signals, including the data bus, are controlled according to their related control timing value and the CYCLETIME counter status.

Wait monitoring is supported for all configurations except for GPMC_CONFIG1_i[19-18] WAITMONITORINGTIME = 0 (where i = 0 to 3) for write bursts with a clock divider of 1 or 2 (GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER field equal to 0 or 1, respectively).

6.1.3.4.8.3.6 WAIT With NAND Device

For details about the use of the wait pin for communication with a NAND flash external device, see Section 6.1.3.4.12.2.

6.1.3.4.8.3.7 Idle Cycle Control Between Successive Accesses

6.1.3.4.8.3.7.1 Bus Turnaround (BUSTURNAROUND)

To prevent data-bus contention, an access that follows a read access to a slow memory/device must be delayed (in other words, control the $\overline{CS}$/$\overline{OE}$ de-assertion to data bus in high-impedance delay).

The bus turnaround is a time-out counter starting after $\overline{CS}$ or $\overline{OE}$ de-assertion time, whichever occurs first, and delays the next access start-cycle time. The counter is programmed through the GPMC_CONFIG6_i[3-0] BUSTURNAROUND bit field (where i = 0 to 7).

After a read access to a chip-select with a non zero BUSTURNAROUND, the next access is delayed until the BUSTURNAROUND delay completes, if the next access is one of the following:

- A write access to any chip-select (same or different from the chip-select data was read from)
- A read access to a different chip-select from the chip-select data was read access from
- A read or write access to a chip-select associated with an address/data-multiplexed device

Bus keeping starts after bus turnaround completion so that DIR changes from IN to OUT after bus turnaround. The bus will not have enough time to go into high-impedance even though it could be driven with the same value before bus turnaround timing.

BUSTURNAROUND delay runs in parallel with GPMC_CONFIG6_i[3-0] CYCLE2CYCLEDELAY delays. It should be noted that BUSTURNAROUND is a timing parameter for the ending chip-select access while CYCLE2CYCLEDELAY is a timing parameter for the following chip-select access. The effective minimum delay between successive accesses is driven by these delay timing parameters and by the access type of the following access. See Figure 6-10 to Figure 6-12.

Another way to prevent bus contention is to define an earlier $\overline{CS}$ or $\overline{OE}$ deassertion time for slow devices or to extend the value of RDCYCLETIME. Doing this prevents bus contention, but affects all accesses of this specific chip-select.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 6-10. Read to Read for an Address-Data Multiplexed Device, On Different CS, Without Bus Turnaround ($\overline{CS0}$ Attached to Fast Device)

Figure 6-11. Read to Read / Write for an Address-Data Multiplexed Device, On Different CS, With Bus Turnaround

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 6-12. Read to Read / Write for a Address-Data or AAD-Multiplexed Device, On Same CS, With Bus Turnaround

6.1.3.4.8.3.7.2 Idle Cycles Between Accesses to Same Chip-Select (CYCLE2CYCLESAMECSEN, CYCLE2CYCLEDELAY)

Some devices require a minimum chip-select signal inactive time between accesses. The GPMC_CONFIG6_i[7] CYCLE2CYCLESAMECSEN bit (i = 0 to 7) enables insertion of a minimum number of GPMC_FCLK cycles, defined by the GPMC_CONFIG6_i[11-8] CYCLE2CYCLEDELAY field, between successive accesses of any type (read or write) to the same chip-select.

If CYCLE2CYCLESAMECSEN is enabled, any subsequent access to the same chip-select is delayed until its CYCLE2CYCLEDELAY completes. The CYCLE2CYCLEDELAY counter starts when CSRDOFFTIME/CSWROFFTIME completes.

The same applies to successive accesses occurring during 32-bit word or burst accesses split into successive single accesses when the single-access mode is used (GPMC_CONFIG1_i[30] READMULTIPLE = 0 or GPMC_CONFIG1_i[28] WRITEMULTIPLE = 0).

All control signals are kept in their default states during these idle GPMC_FCLK cycles. This prevents back-to-back accesses to the same chip-select without idle cycles between accesses.

6.1.3.4.8.3.7.3 Idle Cycles Between Accesses to Different Chip-Select (CYCLE2CYCLEDIFFCSEN, CYCLE2CYCLEDELAY)

Because of the pipelined behavior of the system, successive accesses to different chip-selects can occur back-to-back with no idle cycles between accesses. Depending on the control signals ($\overline{CS}$, $\overline{ADV}$_ALE, $\overline{BE0}$_CLE, $\overline{OE}$_$\overline{RE}$, $\overline{WE}$) assertion and de-assertion timing parameters and on the IC timing parameters, some control signals assertion times may overlap between the successive accesses to different CS. Similarly, some control signals ($\overline{WE}$, $\overline{OE}$_$\overline{RE}$) may not respect required transition times.

To work around the overlapping and to observe the required control-signal transitions, a minimum of CYCLE2CYCLEDELAY inactive cycles is inserted between the access being initiated to this chip-select and the previous access ending for a different chip-select. This applies to any type of access (read or write).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

If GPMC_CONFIG6_i[6] CYCLE2CYCLEDIFFCSEN is enabled, the chip-select access is delayed until CYCLE2CYCLEDELAY cycles have expired since the end of a previous access to a different chip-select. CYCLE2CYCLEDELAY count starts at CSRDOFFTIME/CSWROFFTIME completion. All control signals are kept inactive during the idle GPMC_FCLK cycles.

CYCLE2CYCLESAMECSEN and CYCLE2CYCLEDIFFCSEN should be set in registers to respectively get idle cycles inserted between accesses on this chip-select and after accesses to a different chip-select.

The CYCLE2CYCLEDELAY delay runs in parallel with the BUSTURNAROUND delay. It should be noted that BUSTURNAROUND is a timing parameter defined for the ending chip-select access, whereas CYCLE2CYCLEDELAY is a timing parameter defined for the starting chip-select access. The effective minimum delay between successive accesses is based on the larger delay timing parameter and on access type combination, since bus turnaround does not apply to all access types. See Section 6.1.3.4.8.3.7.1 for more details on bus turnaround.

Table 6-14 describes the configuration required for idle cycle insertion.

Table 6-14. Idle Cycle Insertion Configuration

First Access Type	BUSTURN AROUND Timing Parameter	Second Access Type	Chip-Select	Addr/Data Multiplexed	CYCLE2 CYCLE SAMECSEN Parameter	CYCLE2 CYCLE DIFFCSEN Parameter	Idle Cycle Insertion Between the Two Accesses
R/W	0	R/W	Any	Any	0	x	No idle cycles are inserted if the two accesses are well pipelined.
R	>0	R	Same	Nonmuxed	x	0	No idle cycles are inserted if the two accesses are well pipelined.
R	>0	R	Different	Nonmuxed	0	0	BUSTURNAROUND cycles are inserted.
R	>0	R/W	Any	Muxed	0	0	BUSTURNAROUND cycles are inserted.
R	>0	W	Any	Any	0	0	BUSTURNAROUND cycles are inserted.
W	>0	R/W	Any	Any	0	0	No idle cycles are inserted if the two accesses are well pipelined.
R/W	0	R/W	Same	Any	1	x	CYCLE2CYCLEDELAY cycles are inserted.
R/W	0	R/W	Different	Any	x	1	CYCLE2CYCLEDELAY cycles are inserted.
R/W	>0	R/W	Same	Any	1	x	CYCLE2CYCLEDELAY cycles are inserted. If BTA idle cycles already apply on these two back-to-back accesses, the effective delay is max (BUSTURNAROUND, CYCLE2CYCLEDELAY).
R/W	>0	R/W	Different	Any	x	1	CYCLE2CYCLEDELAY cycles are inserted. If BTA idle cycles already apply on these two back-to-back accesses, the effective delay is maximum (BUSTURNAROUND, CYCLE2CYCLEDELAY).

6.1.3.4.8.3.8 Slow Device Support (TIMEPARAGRANULARITY Parameter)

All access-timing parameters can be multiplied by 2 by setting the GPMC_CONFIG1_i[4] TIMEPARAGRANULARITY bit (where i stands for the GPMC chip-select value, i = 0 to 7). Increasing all access timing parameters allows support of slow devices.

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.8.3.9 GPMC_DIR Pin

The GPMC_DIR pin is used to control I/O direction on the GPMC data bus GPMC_D[15-0]. Depending on top-level pad multiplexing, this signal can be output and used externally to the device, if required. The GPMC_DIR pin is low during transmit (OUT) and high during receive (IN).

For write accesses, the GPMC_DIR pin stays OUT from start-cycle time to end-cycle time.

For read accesses, the GPMC_DIR pin goes from OUT to IN at $\overline{OE}$ assertion time and stays IN until:

- BUSTURNAROUND is enabled
  - The GPMC_DIR pin goes from IN to OUT at end-cycle time plus programmable bus turnaround time.
- BUSTURNAROUND is disabled
  - After an asynchronous read access, the GPMC_DIR pin goes from IN to OUT at RDACCESSTIME + 1 GPMC_FCLK cycle or when RDCYCLETIME completes, whichever occurs last.
  - After a synchronous read access, the GPMC_DIR pin goes from IN to OUT at RDACCESSTIME + 2 GPMC_FCLK cycles or when RDCYCLETIME completes, whichever occurs last.

Because of the bus-keeping feature of the GPMC, after a read or write access and with no other accesses pending, the default value of the GPMC_DIR pin is OUT (see Section 6.1.3.4.9.10). In nonmultiplexed devices, the GPMC_DIR pin stays IN between two successive read accesses to prevent unnecessary toggling.

6.1.3.4.8.3.10 Reset

No reset signal is sent to the external memory device by the GPMC. For more information about external-device reset, see Power, Reset, and Clock Management.

The PRCM module provides an input pin, global_rst_n, to the GPMC:

- The global_rst_n pin is activated during device warm reset and cold reset.
- The global_rst_n pin initializes the internal state-machine and the internal configuration registers.

6.1.3.4.8.3.11 Write Protect Signal ($\overline{WP}$)

When connected to the attached memory device, the write protect signal can enable or disable the lockdown function of the attached memory. The $\overline{GPMC_WP}$ output pin value is controlled through the GPMC_CONFIG[4] WRITEPROTECT bit, which is common to all CS.

6.1.3.4.8.3.12 Byte Enable ($\overline{BE1}/\overline{BE0}$)

Byte enable signals ($\overline{BE1}/\overline{BE0}$) are:

- Valid (asserted or nonasserted according to the incoming system request) from access start to access completion for asynchronous and synchronous single accesses
- Asserted low from access start to access completion for asynchronous and synchronous multiple read accesses
- Valid (asserted or nonasserted, according to the incoming system request) synchronously to each written data for synchronous multiple write accesses

6.1.3.4.8.4 Error Handling

When an error occurs in the GPMC, the error information is stored in the GPMC_ERR_TYPE register and the address of the illegal access is stored in the GPMC_ERR_ADDRESS register. The GPMC keeps only the first error abort information until the GPMC_ERR_TYPE register is reset. Subsequent accesses that cause errors are not logged until the error is cleared by hardware with the GPMC_ERR_TYPE[0]ERRORVALID bit.

- ERRORNOTSUPPADD occurs when an incoming system request address decoding does not match any valid chip-select region, or if two chip-select regions are defined as overlapped, or if a register file access is tried outside the valid address range of 1KB.

Copyright © 2011, Texas Instruments Incorporated

- ERRORNOTSUPPMCMD occurs when an unsupported command request is decoded at the L3 Slow interconnect interface
- ERRORTIMEOUT: A time-out mechanism prevents the system from hanging. The start value of the

9-bit time-out counter is defined in the GPMC_TIMEOUT_CONTROL register and enabled with the GPMC_TIMEOUT_CONTROL[0] TIMEOUTENABLE bit. When enabled, the counter starts at start-cycle time until it reaches 0 and data is not responded to from memory, and then a time-out error occurs. When data are sent from memory, this counter is reset to its start value. With multiple accesses (asynchronous page mode or synchronous burst mode), the counter is reset to its start value for each data access within the burst.

The GPMC does not generate interrupts on these errors. True abort to the MPU or interrupt generation is handled at the interconnect level.

6.1.3.4.9 Timing Setting

The GPMC offers the maximum flexibility to support various access protocols. Most of the timing parameters of the protocol access used by the GPMC to communicate with attached memories or devices are programmable on a chip-select basis. Assertion and deassertion times of control signals are defined to match the attached memory or device timing specifications and to get maximum performance during accesses. For more information on GPMC_CLK and GPMC_FCLK see Section 6.1.3.4.9.6.

In the following sections, the start access time refer to the time at which the access begins.

6.1.3.4.9.1 Read Cycle Time and Write Cycle Time (RDCYCLETIME / WRCYCLETIME)

The GPMC_CONFIG5_i[4-0] RDCYCLETIME and GPMC_CONFIG5_i[12-8] WRCYCLETIME fields (i = 0 to 7) define the address bus and byte enables valid times for read and write accesses. To ensure a correct duty cycle of GPMC_CLK between accesses, RDCYCLETIME and WRCYCLETIME are expressed in GPMC_FCLK cycles and must be multiples of the GPMC_CLK cycle. RDCYCLETIME and WRCYCLETIME bit fields can be set with a granularity of 1 or 2 throught GPMC_CONFIG1_i[4] TIMEPARAGRANULARITY.

When either RDCYCLETIME or WRCYCLETIME completes, if they are not already deasserted, all control signals ($\overline{CS}$, $\overline{ADV}$_ALE, $\overline{OE}$_$\overline{RE}$, $\overline{WE}$, and $\overline{BE0}$_CLE) are deasserted to their reset values, regardless of their deassertion time parameters.

An exception to this forced deassertion occurs when a pipelined request to the same chip-select or to a different chip-select is pending. In such a case, it is not necessary to deassert a control signal with deassertion time parameters equal to the cycle-time parameter. This exception to forced deassertion prevents any unnecessary glitches. This requirement also applies to BE signals, thus avoiding an unnecessary BE glitch transition when pipelining requests.

If no inactive cycles are required between successive accesses to the same or to a different chip-select (GPMC_CONFIG6_i[7] CYCLE2CYCLESAMECSEN = 0 or GPMC_CONFIG6_i[6] CYCLE2CYCLEDIFFCSEN = 0, where i = 0 to 3), and if assertion-time parameters associated with the pipelined access are equal to 0, asserted control signals ($\overline{CS}$, $\overline{ADV}$_ALE, $\overline{BE0}$_CLE, $\overline{WE}$, and $\overline{OE}$_$\overline{RE}$) are kept asserted. This applies to any read/write to read/write access combination.

If inactive cycles are inserted between successive accesses, that is, CYCLE2CYCLESAMECSEN = 1 or CYCLE2CYCLEDIFFCSEN = 1, the control signals are forced to their respective default reset values for the number of GPMC_FCLK cycles defined in CYCLE2CYCLEDELAY.

6.1.3.4.9.2 $\overline{CS}$: Chip-Select Signal Control Assertion/Deassertion Time (CSONTIME / CSRDOFFTIME / CSWROFFTIME / CSEXTRADELAY)

The GPMC_CONFIG2_i[3-0] CSONTIME field (where i = 0 to 7) defines the $\overline{CS}$ signal-assertion time relative to the start access time. It is common for read and write accesses.

The GPMC_CONFIG2_i[12-8] CSRDOFFTIME (read access) and GPMC_CONFIG2_i[20-16] CSWROFFTIME (write access) bit fields define the $\overline{CS}$ signal deassertion time relative to start access time.

Copyright © 2011, Texas Instruments Incorporated

CSONTIME, CSRDOFFTIME and CSWROFFTIME parameters are applicable to synchronous and asynchronous modes. CSONTIME can be used to control an address and byte enable setup time before chip-select assertion. CSRDOFFTIME and CSWROFFTIME can be used to control an address and byte enable hold time after chip-select deassertion.

$\overline{CS}$ signal transitions as controlled through CSONTIME, CSRDOFFTIME, and CSWROFFTIME can be delayed by half a GPMC_FCLK period by enabling the GPMC_CONFIG2_i[7] CSEXTRADELAY bit. This half of a GPMC_FCLK period provides more granularity on the $\overline{CS}$ assertion and deassertion time to guarantee proper setup and hold time relative to GPMC_CLK. CSEXTRADELAY is especially useful in configurations where GPMC_CLK and GPMC_FCLK have the same frequency, but can be used for all GPMC configurations. If enabled, CSEXTRADELAY applies to all parameters controlling $\overline{CS}$ transitions.

The CSEXTRADELAY bit must be used carefully to avoid control-signal overlap between successive accesses to different chip-selects. This implies the need to program the RDCYCLETIME and WRCYCLETIME bit fields to be greater than the $\overline{CS}$ signal-deassertion time, including the extra half-GPMC_FCLK-period delay.

6.1.3.4.9.3 $\overline{ADV}$/ALE: Address Valid/Address Latch Enable Signal Control Assertion/Deassertion Time (ADVONTIME / ADVRDOFFTIME / ADVWROFFTIME / ADVEXTRADELAY/ADVAADMUXONTIME/ADVAADMUXRDOFFTIME/ADVAADMUXWROFFTIME)

The GPMC_CONFIG3_i[3-0] ADVONTIME field (where i = 0 to 7) defines the $\overline{ADV}$_ALE signal-assertion time relative to start access time. It is common to read and write accesses.

The GPMC_CONFIG3_i[12-8] ADVRDOFFTIME (read access) and GPMC_CONFIG3_i[20-16] ADVWROFFTIME (write access) bit fields define the $\overline{ADV}$_ALE signal-deassertion time relative to start access time.

ADVONTIME can be used to control an address and byte enable valid setup time control before $\overline{ADV}$_ALE assertion. ADVRDOFFTIME and ADVWROFFTIME can be used to control an address and byte enable valid hold time control after $\overline{ADV}$_ALE de-assertion. ADVRDOFFTIME and ADVWROFFTIME are applicable to both synchronous and asynchronous modes.

$\overline{ADV}$_ALE signal transitions as controlled through ADVONTIME, ADVRDOFFTIME, and ADVWROFFTIME can be delayed by half a GPMC_FCLK period by enabling the GPMC_CONFIG3_i[7] ADVEXTRADELAY bit. This half of a GPMC_FCLK period provides more granularity on $\overline{ADV}$_ALE assertion and deassertion time to assure proper setup and hold time relative to GPMC_CLK. The ADVEXTRADELAY configuration parameter is especially useful in configurations where GPMC_CLK and GPMC_FCLK have the same frequency, but can be used for all GPMC configurations. If enabled, ADVEXTRADELAY applies to all parameters controlling $\overline{ADV}$_ALE transitions.

ADVEXTRADELAY must be used carefully to avoid control-signal overlap between successive accesses to different chip-selects. This implies the need to program the RDCYCLETIME and WRCYCLETIME bit fields to be greater than $\overline{ADV}$_ALE signal-deassertion time, including the extra half-GPMC_FCLK-period delay.

The GPMC_CONFIG3_i[6-4] ADVAADMUXONTIME, GPMC_CONFIG3_i[26-24] ADVAADMUXRDOFFTIME, and GPMC_CONFIG3_i[30-28] ADVAADMUXWROFFTIME parameters have the same functions as ADVONTIME, ADVRDOFFTIME, and ADVWROFFTIME, but apply to the first address phase in the AAD-multiplexed protocol. It is the user responsibility to make sure ADVAADMUXxxOFFTIME is programmed to a value lower than or equal to ADVxxOFFTIME. Functionality in AAD-mux mode is undefined if the settings do not comply with this requirement. ADVAADMUXxxOFFTIME can be programmed to the same value as ADVONTIME if no high $\overline{ADV}$ pulse is needed between the two AAD-mux address phases, which is the typical case in synchronous mode. In this configuration, $\overline{ADV}$ is kept low until it reaches the correct ADVxxOFFTIME.

See Section 6.1.3.4.12 for more details on ADVONTIME, ADVRDOFFTIME, ADVWROFFTIME, and ADVAADMUXRDOFFTIME, ADVAADMUXWROFFTIME usage for CLE and ALE (Command / Address Latch Enable) usage for a NAND Flash interface.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.9.4 $\overline{OE}$/$\overline{RE}$: Output Enable / Read Enable Signal Control Assertion / Deassertion Time (OEONTIME / OEOFFTIME / OEEXTRADELAY / OEAADMUXONTIME / OEAADMUXOFFTIME)

The GPMC_CONFIG4_i[3-0] OEONTIME field (where i = 0 to 7) defines the $\overline{OE}$_$\overline{RE}$ signal assertion time relative to start access time. It is applicable only to read accesses.

The GPMC_CONFIG4_i[12-8] OEOFFTIME field defines the $\overline{OE}$_$\overline{RE}$ signal deassertion time relative to start access time. It is applicable only to read accesses. $\overline{OE}$_$\overline{RE}$ is not asserted during a write cycle.

OEONTIME, OEOFFTIME, OEAADMUXONTIME and OEAADMUXOFFTIME parameters are applicable to synchronous and asynchronous modes. OEONTIME can be used to control an address and byte enable valid setup time control before $\overline{OE}$_$\overline{RE}$ assertion. OEOFFTIME can be used to control an address and byte enable valid hold time control after $\overline{OE}$_$\overline{RE}$ assertion.

OEAADMUXONTIME and OEAADMUXOFFTIME parameters have the same functions as OEONTIME and OEOFFTIME, but apply to the first OE assertion in the AAD-multiplexed protocol for a read phase, or to the only OE assertion for a write phase. It is the user responsibility to make sure OEAADMUXOFFTIME is programmed to a value lower than OEONTIME. Functionality in AAD-mux mode is undefined if the settings do not comply with this requirement. OEAADMUXOFFTIME shall never be equal to OEONTIME because the AAD-mux protocol requires a second address phase with the $\overline{OE}$ signal de-asserted before $\overline{OE}$ can be asserted again to define a read command.

The $\overline{OE}$_$\overline{RE}$ signal transitions as controlled through OEONTIME, OEOFFTIME, OEAADMUXONTIME and OEAADMUXOFFTIME can be delayed by half a GPMC_FCLK period by enabling the GPMC_CONFIG4_i[7] OEEXTRADELAY bit. This half of a GPMC_FCLK period provides more granularity on $\overline{OE}$_$\overline{RE}$ assertion and deassertion time to assure proper setup and hold time relative to GPMC_CLK. If enabled, OEEXTRADELAY applies to all parameters controlling $\overline{OE}$_$\overline{RE}$ transitions.

OEEXTRADELAY must be used carefully, to avoid control-signal overlap between successive accesses to different chip-selects. This implies the need to program RDCYCLETIME and WRCYCLETIME to be greater than $\overline{OE}$_$\overline{RE}$ signal-deassertion time, including the extra half-GPMC_FCLK-period delay.

When the GPMC generates a read access to an address-/data-multiplexed device, it drives the address bus until $\overline{OE}$ assertion time.

6.1.3.4.9.5 $\overline{WE}$: Write Enable Signal Control Assertion / Deassertion Time (WEONTIME / WEOFFTIME / WEEXTRADELAY)

The GPMC_CONFIG4_i[19-16] WEONTIME field (where i = 0 to 3) defines the $\overline{WE}$ signal-assertion time relative to start access time. The GPMC_CONFIG4_i[28-24] WEOFFTIME field defines the $\overline{WE}$ signal-deassertion time relative to start access time. These bit fields only apply to write accesses. $\overline{WE}$ is not asserted during a read cycle.

WEONTIME can be used to control an address and byte enable valid setup time control before $\overline{WE}$ assertion. WEOFFTIME can be used to control an address and byte enable valid hold time control after $\overline{WE}$ assertion.

$\overline{WE}$ signal transitions as controlled through WEONTIME, and WEOFFTIME can be delayed by half a GPMC_FCLK period by enabling the GPMC_CONFIG4_i[23] WEEXTRADELAY bit. This half of a GPMC_FCLK period provides more granularity on $\overline{WE}$ assertion and deassertion time to guaranty proper setup and hold time relative to GPMC_CLK. If enabled, WEEXTRADELAY applies to all parameters controlling $\overline{WE}$ transitions.

The WEEXTRADELAY bit must be used carefully to avoid control-signal overlap between successive accesses to different chip-selects. This implies the need to program the WRCYCLETIME bit field to be greater than the $\overline{WE}$ signal-deassertion time, including the extra half-GPMC_FCLK-period delay.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.9.6 GPMC_CLK

GPMC_CLK is the external clock provided to the attached synchronous memory or device.

- The GPMC_CLK clock frequency is the GPMC_FCLK functional clock frequency divided by 1, 2, 3, or 4, depending on the GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER bit field (where i = 0 to 7), with a guaranteed 50-percent duty cycle.
- The GPMC_CLK clock is only activated when the access in progress is defined as synchronous (read or write access).
- The GPMC_CONFIG1_i[26-25] CLKACTIVATIONTIME field (i = 0 to 7) defines the number of GPMC_FCLK cycles from start access time to GPMC_CLK activation.
- The GPMC_CLK clock is stopped when cycle time completes and is asserted low between accesses.
- The GPMC_CLK clock is kept low when access is defined as asynchronous.

When cycle time completes, the GPMC_CLK may be high because of the GPMCFCLKDIVIDER bit field. To ensure correct stoppage of the GPMC_CLK clock within the 50-percent required duty cycle, it is the user's responsibility to extend the RDCYCLETIME or WRCYCLETIME value.

To ensure a correct external clock cycle, the following rules must be applied:

- (RDCYCLETIME CLKACTIVATIONTIME) must be a multiple of (GPMCFCLKDIVIDER + 1).
- The PAGEBURSTACCESSTIME value must be a multiple of (GPMCFCLKDIVIDER + 1).

6.1.3.4.9.7 GPMC_CLK and Control Signals Setup and Hold

Control-signal transition (assertion and deassertion) setup and hold values with respect to the GPMC_CLK edge can be controlled in the following ways:

- For the GPMC_CLK signal, the GPMC_CONFIG1_i[26-25] CLKACTIVATIONTIME field (i = 0 to 7) allows setup and hold control of control-signal assertion time.
- The use of a divided GPMC_CLK allows setup and hold control of control-signal assertion and deassertion times.
- When GPMC_CLK runs at the GPMC_FCLK frequency so that GPMC_CLK edge and control-signal transitions refer to the same GPMC_FCLK edge, the control-signal transitions can be delayed by half of a GPMC_FCLK period to provide minimum setup and hold times. This half-GPMC_FCLK delay is enabled with the CSEXTRADELAY, ADVEXTRADELAY, OEEXTRADELAY, or WEEXTRADELAY parameter. This delay must be used carefully to prevent control-signal overlap between successive accesses to different chip-selects. This implies that the RDCYCLETIME and WRCYCLETIME are greater than the last control-signal deassertion time, including the extra half-GPMC_FCLK cycle.

6.1.3.4.9.8 Access Time (RDACCESSTIME / WRACCESSTIME)

The read access time and write access time durations can be programmed independently through GPMC_CONFIG5_i[20-16] RDACCESSTIME and GPMC_CONFIG6_i[28-24] WRACCESSTIME (i = 0 to 7). This allows $\overline{OE}$ and GPMC data capture timing parameters to be independent of $\overline{WE}$ and memory device data capture timing parameters. RDACCESSTIME and WRACCESSTIME bit fields can be set with a granularity of 1 or 2 throught GPMC_CONFIG1_i[4] TIMEPARAGRANULARITY.

6.1.3.4.9.8.1 Access Time on Read Access

In asynchronous read mode, for single and paged accesses, GPMC_CONFIG5_i[[20-16] RDACCESSTIME field (i = 0 to 7) defines the number of GPMC_FCLK cycles from start access time to the GPMC_FCLK rising edge used for the first data capture. RDACCESSTIME must be programmed to the rounded greater value (in GPMC_FCLK cycles) of the read access time of the attached memory device.

In synchronous read mode, for single or burst accesses, RDACCESSTIME defines the number of GPMC_FCLK cycles from start access time to the GPMC_FCLK rising edge corresponding to the GPMC_CLK rising edge used for the first data capture.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

GPMC_CLK which is sent to the memory device for synchronization with the GPMC controller, is internally retimed to correctly latch the returned data. GPMC_CONFIG5_i[4-0] RDCYCLETIME must be greater than RDACCESSTIME in order to let the GPMC latch the last return data using the internally retimed GPMC_CLK.

The external WAIT signal can be used in conjunction with RDACCESSTIME to control the effective GPMC data-capture GPMC_FCLK edge on read access in both asynchronous mode and synchronous mode. For details about wait monitoring, see Section 6.1.3.4.8.1.

6.1.3.4.9.8.2 Access Time on Write Access

In asynchronous write mode, the GPMC_CONFIG6_i[[28-24] WRACCESSTIME timing parameter is not used to define the effective write access time. Instead, it is used as a WAIT invalid timing window, and must be set to a correct value so that the gpmc_wait pin is at a valid state two GPMC_CLK cycles before WRACCESSTIME completes. For details about wait monitoring, see Section 6.1.3.4.8.1.

In synchronous write mode , for single or burst accesses, WRACCESSTIME defines the number of GPMC_FCLK cycles from start access time to the GPMC_CLK rising edge used by the memory device for the first data capture.

The external WAIT signal can be used in conjunction with WRACCESSTIME to control the effective memory device data capture GPMC_CLK edge for a synchronous write access. For details about wait monitoring, see Section 6.1.3.4.8.1.

6.1.3.4.9.9 Page Burst Access Time (PAGEBURSTACCESSTIME)

GPMC_CONFIG5_i[27-24] PAGEBURSTACCESSTIME bit field (i = 0 to 7) can be set with a granularity of 1 or 2 throught the GPMC_CONFIG1_i[[4] TIMEPARAGRANULARITY.

6.1.3.4.9.9.1 Page Burst Access Time on Read Access

In asynchronous page read mode, the delay between successive word captures in a page is controlled through the PAGEBURSTACCESSTIME bit field. The PAGEBURSTACCESSTIME parameter must be programmed to the rounded greater value (in GPMC_FCLK cycles) of the read access time of the attached device.

In synchronous burst read mode, the delay between successive word captures in a burst is controlled through the PAGEBURSTACCESSTIME field.

The external WAIT signal can be used in conjunction with PAGEBURSTACCESSTIME to control the effective GPMC data capture GPMC_FCLK edge on read access. For details about wait monitoring, see Section 6.1.3.4.8.1.

6.1.3.4.9.9.2 Page Burst Access Time on Write Access

Asynchronous page write mode is not supported. PAGEBURSTACCESSTIME is irrelevant in this case.

In synchronous burst write mode, PAGEBURSTACCESSTIME controls the delay between successive memory device word captures in a burst.

The external WAIT signal can be used in conjunction with PAGEBURSTACCESSTIME to control the effective memory-device data capture GPMC_CLK edge in synchronous write mode. For details about wait monitoring, see Section 6.1.3.4.8.1.

6.1.3.4.9.10 Bus Keeping Support

At the end-cycle time of a read access, if no other access is pending, the GPMC drives the bus with the last data read after RDCYCLETIME completion time to prevent bus floating and reduce power consumption.

After a write access, if no other access is pending, the GPMC keeps driving the data bus after WRCYCLETIME completes with the same data to prevent bus floating and power consumption.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.10 NOR Access Description

For each chip-select configuration, the read access can be specified as either asynchronous or synchronous access through the GPMC_CONFIG1_i[29] READTYPE bit (i = 0 to 7). For each chip-select configuration, the write access can be specified as either synchronous or asynchronous access through the GPMC_CONFIG1_i[27] WRITETYPE bit (i = 0 to 7).

Asynchronous and synchronous read and write access time and related control signals are controlled through timing parameters that refer to GPMC_FCLK. The primary difference of synchronous mode is the availability of a configurable clock interface (GPMC_CLK) to control the external device. Synchronous mode also affects data-capture and wait-pin monitoring schemes in read access.

For details about asynchronous and synchronous access, see the descriptions of GPMC_CLK, RdAccessTime, WrAccessTime, and wait-pin monitoring.

For more information about timing-parameter settings, see the sample timing diagrams in this chapter.

The address bus and $\overline{\text{BE[1:0]}}$ are fixed for the duration of a synchronous burst read access, but they are updated for each beat of an asynchronous page-read access.

6.1.3.4.10.1 Asynchronous Access Description

This section describes:

- Asynchronous single read operation on an address/data multiplexed device
- Asynchronous single write operation on an address/data-multiplexed device
- Asynchronous single read operation on an AAD-multiplexed device
- Asynchronous single write operation on an AAD-multiplexed device
- Asynchronous multiple (page) read operation on a non-multiplexed device

In asynchronous operations GPMC_CLK is not provided outside the GPMC and is kept low.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.10.1.1 Access on Address/Data Multiplexed Devices

6.1.3.4.10.1.1.1 Asynchronous Single-Read Operation on an Address/Data Multiplexed Device

Figure 6-13 shows an asynchronous single read operation on an address/data-multiplexed device.

Figure 6-13. Asynchronous Single Read Operation on an Address/Data Multiplexed Device

6.1.3.4.10.1.1.2 Asynchronous Single Read on an Address/Data-Multiplexed Device

See Section 6.1.3.10.1 for formulas to calculate timing parameters.

Table 6-45 lists the timing bit fields to set up in order to configure the GPMC in asynchronous single read mode.

When the GPMC generates a read access to an address/data-multiplexed device, it drives the address bus until $\overline{OE}$ assertion time. For details, see Section 6.1.3.4.8.2.3.

Address bits (A[16:1] from a GPMC perspective, A[15:0] from an external device perspective) are placed on the address/data bus, and the remaining address bits GPMC_A[25:16] are placed on the address bus. The address phase ends at $\overline{OE}$ assertion, when the DIR signal goes from OUT to IN.

- Chip-select signal $\overline{CS}$
  - $\overline{CS}$ assertion time is controlled by the GPMC_CONFIG2_i[3-0] CSONTIME field. It controls the address setup time to $\overline{CS}$ assertion.
  - $\overline{CS}$ deassertion time is controlled by the GPMC_CONFIG2_i[12-8] CSRDOFFTIME field. It controls the address hold time from $\overline{CS}$ deassertion
- Address valid signal $\overline{ADV}$
  - $\overline{ADV}$ assertion time is controlled by the GPMC_CONFIG3_i[3-0] ADVONTIME field.
  - $\overline{ADV}$ deassertion time is controlled by the GPMC_CONFIG3_i[[12-8] ADVRDOFFTIME field.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

- Output enable signal $\overline{OE}$
  - $\overline{OE}$ assertion indicates a read cycle.
  - $\overline{OE}$ assertion time is controlled by the GPMC_CONFIG4_i[[3-0] OEONTIME field.
  - OE deassertion time is controlled by the GPMC_CONFIG4_i[[12-8] OEOFFTIME field.
- Read data is latched when RDACCESSTIME completes. Access time is defined in the GPMC_CONFIG5_i[20-16] RDACCESSTIME field.
- Direction signal DIR: DIR goes from OUT to IN at the same time that $\overline{OE}$ is asserted.
- The end of the access is defined by the GPMC_CONFIG5_i[4-0] RDCYCLETIME parameter.

In the GPMC, when a 16-bit wide device is attached to the controller, a 32-bit word write access is split into two 16-bit word write accesses. For more information about GPMC access size and type adaptation, see Section 6.1.3.4.10.5. Between two successive accesses, if an $\overline{CS}$ pulse is needed:

- The GPMC_CONFIG6_i[[11-8] CYCLE2CYCLEDELAY field can be programmed with GPMC_CONFIG6_i[[7] CYCLE2CYCLESAMECSEN enabled.
- The CSWROFFTIME and CSONTIME parameters also allow a chip-select pulse, but this affects all other types of access.

Figure 6-14 shows two asynchronous single-read accesses on an address/data-multiplexed devices.

Figure 6-14. Two Asynchronous Single Read Accesses on an Address/Data Multiplexed Device (32-Bit Read Split Into 2 × 16-Bit Read)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.10.1.1.3 Asynchronous Single Write Operation on an Address/Data-Multiplexed Device

Figure 6-15 shows an asynchronous single write operation on an address/data-multiplexed device.

Figure 6-15. Asynchronous Single Write on an Address/Data-Multiplexed Device

6.1.3.4.10.1.1.4 Asynchronous Single Write on an Address/Data-Multiplexed Device

See Section 6.1.3.10.1 for formulas to calculate timing parameters.

Table 6-45 lists the timing bit fields to set up in order to configure the GPMC in asynchronous single write mode. When the GPMC generates a write access to an address/data-multiplexed device, it drives the address bus until $\overline{WE}$ assertion time. For more information, see Section 6.1.3.4.8.2.3.

The $\overline{CS}$ and $\overline{ADV}$ signals are controlled in the same way as for asynchronous single read operation on an address/data-multiplexed device.

- Write enable signal $\overline{WE}$
  - $\overline{WE}$ assertion indicates a write cycle.
  - $\overline{WE}$ assertion time is controlled by the GPMC_CONFIG4_i[19-16] WEONTIME field.
  - $\overline{WE}$ deassertion time is controlled by the GPMC_CONFIG4_i[28-24] WEOFFTIME field.
- Direction signal DIR: DIR signal is OUT during the entire access.
- The end of the access is defined by the GPMC_CONFIG5_i[12-8] WRCYCLETIME parameter.

Address bits A[16:1] (GPMC point of view) are placed on the address/data bus at the start of cycle time, and the remaining address bits A[26:17] are placed on the address bus.

Data is driven on the address/data bus at a GPMC_CONFIG6_i[19-16] WRDATAONADMUXBUS time.

Write multiple access in asynchronous mode is not supported. If WRITEMULTIPLE is enabled with WRITETYPE as asynchronous, the GPMC processes single asynchronous accesses.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

After a write operation, if no other access (read or write) is pending, the data bus keeps its previous value. See Section 6.1.3.4.9.10.

6.1.3.4.10.1.1.5 Asynchronous Multiple (Page) Write Operation on an Address/Data-Multiplexed Device

Write multiple (page) access in asynchronous mode is not supported for address/data-multiplexed devices. If GPMC_CONFIG1_i[28] WRITEMULTIPLE is enabled (1) with GPMC_CONFIG1_i[27] WRITETYPE as asynchronous (0), the GPMC processes single asynchronous accesses.

For accesses on non-multiplexed devices, see Section 6.1.3.4.10.3.

6.1.3.4.10.1.2 Access on Address/Address/Data (AAD) Multiplexed Devices

6.1.3.4.10.1.2.1 Asynchronous Single Read Operation on an AAD-Multiplexed Device

Figure 6-16 shows an asynchronous single read operation on an AAD-multiplexed device.

Figure 6-16. Asynchronous Single-Read on an AAD-Multiplexed Device

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.10.1.2.2 Asynchronous Single Read on an AAD-Multiplexed Device

See Section 6.1.3.10.1 for formulas to calculate timing parameters.

Table 6-45 lists the timing bit fields to set up in order to configure the GPMC in asynchronous single write mode.

When the GPMC generates a read access to an AAD-multiplexed device, all address bits are driven onto the address/data bus in two separate phases. The first phase is used for the MSB address and is qualified with $\overline{OE}$ driven low. The first address phase ends at the first $\overline{OE}$ deassertion time. The second phase for LSB address is qualified with $\overline{OE}$ driven high. The second address phase ends at the second $\overline{OE}$ assertion time, when the DIR signal goes from OUT to IN.

The $\overline{CS}$ and DIR signals are controlled in the same way as for asynchronous single read operation on an address/data-multiplexed device.

- Address valid signal $\overline{ADV}$. $\overline{ADV}$ is asserted and deasserted twice during a read transaction:
  - $\overline{ADV}$ first assertion time is controlled by the GPMC_CONFIG3_i[6-4] ADVAADMUXONTIME field.
  - $\overline{ADV}$ first deassertion time is controlled by the GPMC_CONFIG3_i[26-24] ADVAADMUXRDOFFTIME field.
  - $\overline{ADV}$ second assertion time is controlled by the GPMC_CONFIG3_i[3-0] ADVONTIME field.
  - $\overline{ADV}$ second deassertion time is controlled by the GPMC_CONFIG3_i[12-8] ADVRDOFFTIME field.
- Output Enable signal $\overline{OE}$. $\overline{OE}$ is asserted and deasserted twice during a read transaction ($\overline{OE}$ second assertion indicates a read cycle):
  - $\overline{OE}$ first assertion time is controlled by the GPMC_CONFIG4_i[6-4] OEAADMUXONTIME field.
  - $\overline{OE}$ first deassertion time is controlled by the GPMC_CONFIG3_i[15-13] OEAADMUXOFFTIME field.
  - $\overline{OE}$ second assertion time is controlled by the GPMC_CONFIG4_i[3-0] OEONTIME field.
  - $\overline{OE}$ second deassertion time is controlled by the GPMC_CONFIG4_i[12-8] OEOFFTIME field.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.10.1.2.3 Asynchronous Single Write Operation on an AAD-Multiplexed Device

Figure 6-17 shows an asynchronous single write operation on an AAD-multiplexed device.

Figure 6-17. Asynchronous Single Write on an AAD-Multiplexed Device

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

See Section 6.1.3.10.1 for formulas to calculate timing parameters.

Table 6-45 lists the timing bit fields to set up to configure the GPMC in asynchronous single write mode.

When the GPMC generates a write access to an AAD-multiplexed device, all address bits are driven onto the address/data bus in two separate phases. The first phase is used for the MSB address and is qualified with $\overline{OE}$ driven low. The second phase for LSB address is qualified with $\overline{OE}$ driven high. The address phase ends at $\overline{WE}$ assertion time.

The $\overline{CS}$, $\overline{WE}$, and DIR signals are controlled in the same way as for asynchronous single write operation on an address/data-multiplexed device.

- Address valid signal $\overline{ADV}$ is asserted and deasserted twice during a write transaction
  - $\overline{ADV}$ first assertion time is controlled by the GPMC_CONFIG3_i[6-4] ADVAADMUXONTIME field.
  - $\overline{ADV}$ first deassertion time is controlled by the GPMC_CONFIG3_i[30-28] ADVAADMUXWROFFTIME field.
  - $\overline{ADV}$ second assertion time is controlled by the GPMC_CONFIG3_i[3-0] ADVONTIME field.
  - $\overline{ADV}$ second deassertion time is controlled by the GPMC_CONFIG3_i[20-16] ADVWROFFTIME field.
- Output Enable signal $\overline{OE}$ is asserted during the address phase of a write transaction
  - $\overline{OE}$ assertion time is controlled by the GPMC_CONFIG4_i[6-4] OEAADMUXONTIME field.
  - $\overline{OE}$ deassertion time is controlled by the GPMC_CONFIG3_i[15-13] OEAADMUXOFFTIME field.

The address bits for the first address phase are driven onto the data bus until $\overline{OE}$ deassertion. Data is driven onto the address/data bus at the clock edge defined by the GPMC_CONFIG6_i[19-16] WRDATAONADMUXBUS parameter.

6.1.3.4.10.1.2.4 Asynchronous Multiple (Page) Read Operation on an AAD-Multiplexed Device

Write multiple (page) access in asynchronous mode is not supported for AAD-multiplexed devices.

If GPMC_CONFIG1_i[28] WRITEMULTIPLE is enabled (1) with GPMC_CONFIG1_i[27] WRITETYPE as asynchronous (0), the GPMC processes single asynchronous accesses.

For accesses on non-multiplexed devices, see Section 6.1.3.4.10.3.

6.1.3.4.10.2 Synchronous Access Description

This section details read and write synchronous accesses on address/data multiplexed. All information in this section can be applied to any type of memory - non-multiplexed, address and data multiplexed or AAD-multiplexed - with a difference limited to the address phase. For accesses on non-multiplexed devices, see Section 6.1.3.4.10.3.

In synchronous operations:

- The GPMC_CLK clock is provided outside the GPMC when accessing the memory device.
- The GPMC_CLK clock is derived from the GPMC_FCLK clock using the GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER field. In the following section, i stands for the chip-select number, i = 0 to 3.
- The GPMC_CONFIG1_i[26-25] CLKACTIVATIONTIME field specifies that the GPMC_CLK is provided outside the GPMC 0, 1, or 2 GPMC_FCLK cycles after start access time until RDCYCLETIME or WRCYCLETIME completion.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.10.2.1 Synchronous Single Read

Figure 6-18 and Figure 6-19 show a synchronous single-read operation with GPMCFCLKDIVIDER equal to 0 and 1, respectively.

Figure 6-18. Synchronous Single Read (GPMCFCLKDIVIDER = 0)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 6-19. Synchronous Single Read (GPMCFCLKDIVIDER = 1)

See Section 6.1.3.10.1 for formulas to calculate timing parameters.

Table 6-45 lists the timing bit fields to set up in order to configure the GPMC in asynchronous single read mode.

When the GPMC generates a read access to an address/data-multiplexed device, it drives the address bus until $\overline{OE}$ assertion time. For details, see Section 6.1.3.4.8.2.3.

- Chip-select signal $\overline{CS}$
  - $\overline{CS}$ assertion time is controlled by the GPMC_CONFIG2_i[3-0] CSONTIME field and ensures address setup time to $\overline{CS}$ assertion.
  - $\overline{CS}$ deassertion time is controlled by the GPMC_CONFIG2_i[12-8] CSRDOFFTIME field and ensures address hold time to $\overline{CS}$ deassertion.
- Address valid signal $\overline{ADV}$
  - $\overline{ADV}$ assertion time is controlled by the GPMC_CONFIG3_i[3-0] ADVONTIME field.
  - $\overline{ADV}$ deassertion time is controlled by the GPMC_CONFIG3_i[12-8] ADVRDOFFTIME field.
- Output enable signal $\overline{OE}$
  - $\overline{OE}$ assertion indicates a read cycle.
  - $\overline{OE}$ assertion time is controlled by the GPMC_CONFIG4_i[3-0] OEONTIME field.
  - $\overline{OE}$ deassertion time is controlled by the GPMC_CONFIG4_i[12-8] OEOFFTIME field.
- Initial latency for the first read data is controlled by GPMC_CONFIG5_i[20-16] RDACCESSTIME or by monitoring the WAIT signal.
- Total access time (GPMC_CONFIG5_i[4-0] RDCYCLETIME) corresponds to RDACCESSTIME plus the address hold time from $\overline{CS}$ deassertion, plus time from RDACCESSTIME to CSRDOFFTIME.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

- Direction signal DIR: DIR goes from OUT to IN at the same time as $\overline{OE}$ assertion.

When the GPMC generates a write access to an AAD-multiplexed device, all address bits are driven onto the address/data bus in two separate phases. The first phase is used for the MSB address and is qualified with $\overline{OE}$ driven low. The second phase for LSB address is qualified with $\overline{OE}$ driven high. The address phase ends at $\overline{WE}$ assertion time.

The $\overline{CS}$ and DIR signals are controlled in the same way as for synchronous single read operation on an address/data-multiplexed device.

- Address valid signal $\overline{ADV}$ is asserted and deasserted twice during a read transaction
  - $\overline{ADV}$ first assertion time is controlled by the GPMC_CONFIG3_i[6-4] ADVAADMUXONTIME field.
  - $\overline{ADV}$ first deassertion time is controlled by the GPMC_CONFIG3_i[26-24] ADVAADMUXRDOFFTIME field.
  - $\overline{ADV}$ second assertion time is controlled by the GPMC_CONFIG3_i[3-0] ADVONTIME field.
  - $\overline{ADV}$ second deassertion time is controlled by the GPMC_CONFIG3_i[12-8] ADVRDOFFTIME field.
- Output Enable signal $\overline{OE}$ is asserted and deasserted twice during a read transaction ($\overline{OE}$ second assertion indicates a read cycle)
  - $\overline{OE}$ first assertion time is controlled by the GPMC_CONFIG4_i[6-4] OEAADMUXONTIME field.
  - $\overline{OE}$ first deassertion time is controlled by the GPMC_CONFIG3_i[15-13] OEAADMUXOFFTIME field.
  - $\overline{OE}$ second assertion time is controlled by the GPMC_CONFIG4_i[3-0] OEONTIME field.
  - $\overline{OE}$ second deassertion time is controlled by the GPMC_CONFIG4_i[12-8] OEOFFTIME field.

After a read operation, if no other access (read or write) is pending, the data bus is driven with the previous read value. See Section 6.1.3.4.9.10.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.10.2.2 Synchronous Multiple (Burst) Read (4-, 8-, 16-Word16 Burst With Wraparound Capability)

Figure 6-20 and Figure 6-21 show a synchronous multiple read operation with GPMCFCLKDivider equal to 0 and 1, respectively.

Figure 6-20. Synchronous Multiple (Burst) Read (GPMCFCLKDIVIDER = 0)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 6-21. Synchronous Multiple (Burst) Read (GPMCFCLKDIVIDER = 1)

When GPMC_CONFIG5_i[20-16] RDACCESSTIME completes, control-signal timings are frozen during the multiple data transactions, corresponding to GPMC_CONFIG5_i[27-24] PAGEBURSTACCESSTIME multiplied by the number of remaining data transactions.

The $\overline{CS}$, $\overline{ADV}$, $\overline{OE}$ and DIR signals are controlled in the same way as for synchronous single read operation. See Section 6.1.3.4.10.2.1.

Initial latency for the first read data is controlled by RDACCESSTIME or by monitoring the WAIT signal. Successive read data are provided by the memory device each one or two GPMC_CLK cycles. The PAGEBURSTACCESSTIME parameter must be set accordingly with GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER and the memory-device internal configuration. Depending on the device page length, the GPMC checks device page crossing during a new burst request and purposely insert initial latency (of RDACCESSTIME) when required.

Total access time GPMC_CONFIG5_i[4-0] RDCYCLETIME corresponds to RDACCESSTIME plus the address hold time from $\overline{CS}$ deassertion. In Figure 6-20, RDCYCLETIME programmed value equals to RDCYCLETIME0 + RDCYCLETIME1.

After a read operation, if no other access (read or write) is pending, the data bus is driven with the previous read value. See Section 6.1.3.4.9.10.

Burst wraparound is enabled through the GPMC_CONFIG1_i[31] WRAPBURST bit and allows a 4-, 8-, or 16-Word16 linear burst access to wrap within its burst-length boundary through GPMC_CONFIG1_i[24-23] ATTACHEDDEVICEPAGELENGTH.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.10.2.3 Synchronous Single Write

Burst write mode is used for synchronous single or burst accesses (see Figure 6-22).

Figure 6-22. Synchronous Single Write on an Address/Data-Multiplexed Device

When the GPMC generates a write access to an address/data-multiplexed device, it drives the data bus (with address bits A[16:1]) until [19:16] WRDATAONADMUXBUS time. First data of the burst is driven on the address/data bus at WRDATAONADMUXBUS time.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.10.2.4 Synchronous Multiple (Burst) Write

Synchronous burst write mode provides synchronous single or consecutive accesses. Figure 6-23 shows a synchronous burst write access when the chip-select is configured in address/data-multiplexed mode.

Figure 6-23. Synchronous Multiple Write (Burst Write) in Address/Data-Multiplexed Mode

Figure 6-24 shows the same synchronous burst write access when the chip-select is configured in address/address/data-multiplexed (AAD-multiplexed) mode.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 6-24. Synchronous Multiple Write (Burst Write) in Address/Address/Data-Multiplexed Mode

The first data of the burst is driven on the A/D bus at GPMC_CONFIG6_i[19:16] WRDATAONADMUXBUS.

When WRACCESSTIME completes, control-signal timings are frozen during the multiple data transactions, corresponding to the GPMC_CONFIG5_i[27-24] PAGEBURSTACCESSTIME multiplied by the number of remaining data transactions.

When the GPMC generates a read access to an address/data-multiplexed device, it drives the address bus until $\overline{OE}$ assertion time. For details, see Section 6.1.3.4.8.2.3.

- Chip-select signal $\overline{CS}$
  - $\overline{CS}$ assertion time is controlled by the GPMC_CONFIG2_i[3-0] CSONTIME field (i = 0 to 7) and ensures address setup time to $\overline{CS}$ assertion.
  - $\overline{CS}$ deassertion time controlled by the GPMC_CONFIG2_i[20-16] CSWROFFTIME field and ensures address hold time to $\overline{CS}$ deassertion.
- Address valid signal $\overline{ADV}$
  - $\overline{ADV}$ assertion time is controlled by the GPMC_CONFIG3_i[3-0] ADVONTIME field.
  - $\overline{ADV}$ deassertion time is controlled by the GPMC_CONFIG3_i[20-16] ADVWROFFTIME field.

Copyright © 2011, Texas Instruments Incorporated

- Write enable signal $\overline{WE}$
  - $\overline{WE}$ assertion indicates a read cycle.
  - $\overline{WE}$ assertion time is controlled by the GPMC_CONFIG4_i[19-16] WEONTIME field.
  - $\overline{WE}$ deassertion time is controlled by the GPMC_CONFIG4_i[28-24] WEOFFTIME field.

  The $\overline{WE}$ falling edge must not be used to control the time when the burst first data is driven in the address/data bus because some new devices require the $\overline{WE}$ signal at low during the address phase.
- Direction signal DIR is OUT during the entire access.

When the GPMC generates a write access to an AAD-multiplexed device, all address bits are driven onto the address/data bus in two separate phases. The first phase is used for the MSB address and is qualified with $\overline{OE}$ driven low. The second phase for LSB address is qualified with $\overline{OE}$ driven high. The address phase ends at $\overline{WE}$ assertion time.

The $\overline{CS}$ and DIR signals are controlled as detailled above.

- Address valid signal $\overline{ADV}$ is asserted and deasserted twice during a read transaction
  - $\overline{ADV}$ first assertion time is controlled by the GPMC_CONFIG3_i[[6-4] ADVAADMUXONTIME field.
  - $\overline{ADV}$ first deassertion time is controlled by the GPMC_CONFIG3_i[[26-24] ADVAADMUXRDOFFTIME field.
  - $\overline{ADV}$ second assertion time is controlled by the GPMC_CONFIG3_i[[3-0] ADVONTIME field.
  - $\overline{ADV}$ second deassertion time is controlled by the GPMC_CONFIG3_i[[12-8] ADVRDOFFTIME field.
- Output Enable signal $\overline{OE}$ is asserted and deasserted twice during a read transaction ($\overline{OE}$ second assertion indicates a read cycle)
  - $\overline{OE}$ first assertion time is controlled by the GPMC_CONFIG4_i[[6-4] OEAADMUXONTIME field.
  - $\overline{OE}$ first deassertion time is controlled by the GPMC_CONFIG4_i[15-13] OEAADMUXOFFTIME field.
  - $\overline{OE}$ second assertion time is controlled by the GPMC_CONFIG4_i[3-0] OEONTIME field.
  - $\overline{OE}$ second deassertion time is controlled by the GPMC_CONFIG4_i[12-8] OEOFFTIME field.

First write data is driven by the GPMC at GPMC_CONFIG6_i[19-16] WRDATAONADMUXBUS, when in address/data mux configuration. The next write data of the burst is driven on the bus at WRACCESSTIME + 1 during GPMC_CONFIG5_i[27-24] PAGEBURSTACCESSTIME GPMC_FCLK cycles. The last data of the synchronous burst write is driven until GPMC_CONFIG5_i[12-8] WRCYCLETIME completes.

- WRACCESSTIME is defined in the GPMC_CONFIG6_i[28-24] register.
- The PAGEBURSTACCESSTIME parameter must be set accordingly with GPMCFCLKDIVIDER and the memory-device internal configuration.

Total access time GPMC_CONFIG5_i[12-8] WRCYCLETIME corresponds to WRACCESSTIME plus the address hold time from $\overline{CS}$ deassertion. In Figure 6-24 the WRCYCLETIME programmed value equals WRCYCLETIME0 + WRCYCLETIME1. WRCYCLETIME0 and WRCYCLETIME1 delays are not actual parameters and are only a graphical representation of the full WRCYCLETIME value.

After a write operation, if no other access (read or write) is pending, the data bus keeps the previous value. See Section 6.1.3.4.9.10.

6.1.3.4.10.3 Asynchronous and Synchronous Accesses in Nonmultiplexed Mode

Page mode is only available in non-multiplexed mode.

- Asynchronous single read operation on a nonmultiplexed device
- Asynchronous single write operation on a nonmultiplexed device
- Asynchronous multiple (page mode) read operation on a nonmultiplexed device
- Synchronous operations on a nonmultiplexed device

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.10.3.1 Asynchronous Single Read Operation on a Nonmultiplexed Device

Figure 6-25 shows an asynchronous single read operation on a nonmultiplexed device.

Figure 6-25. Asynchronous Single Read on an Address/Data-Nonmultiplexed Device

The 27-bit address is driven onto the address bus A[27:1] and the 16-bit data is driven onto the data bus D[15:0].

Read data is latched at GPMC_CONFIG1_5[20-16] RDACCESSTIME completion time. The end of the access is defined by the GPMC_CONFIG1_5[4-0] RDCYCLETIME parameter.

$\overline{CS}$, $\overline{ADV}$, $\overline{OE}$ and DIR signals are controlled in the same way as address/data multiplexed accesses, see Section 6.1.3.4.10.1.1.2.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.10.3.2 Asynchronous Single Write Operation on a Nonmultiplexed Device

Figure 6-26 shows an asynchronous single write operation on a nonmultiplexed device.

Figure 6-26. Asynchronous Single Write on an Address/Data-Nonmultiplexed Device

The 27-bit address is driven onto the address bus A[27:1] and the 16-bit data is driven onto the data bus D[15:0].

$\overline{CS}$, $\overline{ADV}$, $\overline{WE}$ and DIR signals are controlled in the same way as address/data multiplexed accesses, see Section 6.1.3.4.10.1.1.3.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.10.3.3 Asynchronous Multiple (Page Mode) Read Operation on a Nonmultiplexed Device

Figure 6-27 shows an asynchronous multiple read operation on a Nonmultiplexed Device, in which two word32 host read accesses to the GPMC are split into one multiple (page mode of 4 word16) read access to the attached device.

Figure 6-27. Asynchronous Multiple (Page Mode) Read

The WAIT signal is active low.

$\overline{CS}$, $\overline{ADV}$, $\overline{OE}$ and DIR signals are controlled in the same way as address/data multiplexed accesses, see Section 6.1.3.4.10.1.1.2.

When RDACCESSTIME completes, control-signal timings are frozen during the multiple data transactions, corresponding to PAGEBURSTACCESSTIME multiplied by the number of remaining data transactions.

Read data is latched at GPMC_CONFIG5_i[20-16] RDACCESSTIME completion time (i = 0 to 7). The end of the access is defined by the GPMC_CONFIG5_i[4-0] RDCYCLETIME parameter.

During consecutive accesses, the GPMC increments the address after each data read completes.

Delay between successive read data in the page is controlled by the GPMC_CONFIG5_i[27-24] PAGEBURSTACCESSTIME parameter. Depending on the device page length, the GPMC can control device page crossing during a burst request and insert initial RDACCESSTIME latency. Note that page crossing is only possible with a new burst access, meaning a new initial access phase is initiated.

Total access time RDCYCLETIME corresponds to RDACCESSTIME plus the address hold time starting from the $\overline{CS}$ deassertion.

- The read cycle time is defined in the GPMC_CONFIG5_i[4-0] RDCYCLETIME field.
- In Figure 6-27, the RDCYCLETIME programmed value equals RDCYCLETIME0 (before paged accesses) + RDCYCLETIME1 (after paged accesses).

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.10.3.4 Synchronous Operations on a Nonmultiplexed Device

All information for this section is equivalent to similar operations for address/data- or AAD-multiplexed accesses. The only difference resides in the address phase. See Section 6.1.3.7.

6.1.3.4.10.4 Page and Burst Support

Each chip-select can be configured to process system single or burst requests into successive single accesses or asynchronous page/synchronous burst accesses, with appropriate access size adaptation.

Depending on the external device page or burst capability, read and write accesses can be independently configured through the GPMC. The GPMC_CONFIG1_i[30] READMULTIPLE and GPMC_CONFIG1_i[28] WRITMULTIPLE bits (i = 0 to 7) are associated with the READTYPE and WRITETYPE parameters.

- Asynchronous write page mode is not supported.
- 8-bit wide device support is limited to nonburstable devices (READMULTIPLE and WRITEMULTIPLE are ignored).
- Not applicable to NAND device interfacing.

6.1.3.4.10.5 System Burst Versus External Device Burst Support

The device system can issue the following requests to the GPMC:

- Byte, 16-bit word, 32-bit word requests (byte enable controlled). This is always a single request from the interconnect point of view.
- Incrementing fixed-length bursts of two words, four words, and eight words
- Wrapped (critical word access first) fixed-length burst of two, four, or eight words

To process a system request with the optimal protocol, the READMULTIPLE (and READTYPE) and WRITEMULTIPLE (and WRITETYPE) parameters must be set according to the burstable capability (synchronous or asynchronous) of the attached device.

The GPMC access engine issues only fixed-length burst. The maximum length that can be issued is defined per CS by the GPMC_CONFIG1_i[24-23] ATTACHEDDEVICEPAGELENGTH field (i = 0 to 7). When the ATTACHEDDEVICEPAGELENGTH value is less than the system burst request length (including the appropriate access size adaptation according to the device width), the GPMC splits the system burst request into multiple bursts. Within the specified 4-, 8-, or 16-word value, the ATTACHEDDEVICEPAGELENGTH field value must correspond to the maximum-length burst supported by the memory device configured in fixed-length burst mode (as opposed to continuous burst mode).

To get optimal performance from memory devices that natively support 16 Word16-length-wrapping burst capability (critical word access first), the ATTACHEDDEVICEPAGELENGTH parameter must be set to 16 words and the GPMC_CONFIG1_i[31] WRAPBURST bit (i = 0 to 7) must be set to 1. Similarly DEVICEPAGELENGTH is set to 4 and 8 for memories supporting respectively 4 and 8 Word16-length-wrapping burst.

When the memory device does not offer (or is not configured to offer) native 16 Word16-length-wrapping burst, the WRAPBURST parameter must be cleared, and the GPMC access engine emulates the wrapping burst by issuing the appropriate burst sequences according to the ATTACHEDDEVICEPAGELENGTH value.

When the memory device does not support native-wrapping burst, there is usually no difference in behavior between a fixed burst length mode and a continuous burst mode configuration (except for a potential power increase from a memory-speculative data prefetch in a continuous burst read). However, even though continuous burst mode is compatible with GPMC behavior, because the GPMC access engine issues only fixed-length burst and does not benefit from continuous burst mode, it is best to configure the memory device in fixed-length burst mode.

The memory device maximum-length burst (configured in fixed-length burst wrap or nonwrap mode) usually corresponds to the memory device data buffer size. Memory devices with a minimum of 16 half-word buffers are the most appropriate (especially with wrap support), but memory devices with smaller buffer size (4 or 8) are also supported, assuming that the GPMC_CONFIG1_i[24-23] ATTACHEDDEVICEPAGELENGTH field is set accordingly to 4 or 8 words.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

The device system issues only requests with addresses or starting addresses for nonwrapping burst requests; that is, the request size boundary is aligned. In case of an eight-word-wrapping burst, the wrapping address always occurs on the eight-words boundary. As a consequence, all words requested must be available from the memory data buffer when the buffer size is equal to or greater than the ATTACHEDDEVICEPAGELENGTH value. This usually means that data can be read from or written to the buffer at a constant rate (number of cycles between data) without wait states between data accesses. If the memory does not behave this way (nonzero wait state burstable memory), wait-pin monitoring must be enabled to dynamically control data-access completion within the burst.

When the system burst request length is less than the ATTACHEDDEVICEPAGELENGTH value, the GPMC proceeds with the required accesses.

6.1.3.4.11 pSRAM Access Specificities

pSRAM devices are SRAM-pin-compatible low-power memories that contain a self-refreshed DRAM memory array. The GPMC_CONFIG1_i[[11-10] DEVICETYPE field (i = 0 to 3) shall be cleared to 0b00.

The pSRAM devices uses the NOR protocol. It support the following operations:

- Asynchronous single read
- Asynchronous page read
- Asynchronous single write
- Synchronous single read and write
- Synchronous burst read
- Synchronous burst write (not supported by NOR Flash memory)

pSRAM devices must be powered up and initialized in a predefined manner according to the specifications of the attached device.

pSRAM devices can be programmed to use either mode: fixed or variable latency. pSRAM devices can either automatically schedule autorefresh operations, which force the GPMC to use its WAIT signal capability when read or write operations occur during an internal self-refresh operation, or pSRAM devices automatically include the autorefresh operation in the access time. These devices do not require additional WAIT signal capability or a minimum $\overline{CS}$ high pulse width between consecutive accesses to ensure that the correct internal refresh operation is scheduled.

6.1.3.4.12 NAND Access Description

NAND (8-bit and 16-bit) memory devices using a standard NAND asynchronous address/data-multiplexing scheme can be supported on any chip-select with the appropriate asynchronous configuration settings

As for any other type of memory compatible with the GPMC interface, accesses to a chip-select allocated to a NAND device can be interleaved with accesses to chip-selects allocated to other external devices. This interleaved capability limits the system to *chip enable don't care* NAND devices, because the chip-select allocated to the NAND device must be de-asserted if accesses to other chip-selects are requested.

6.1.3.4.12.1 NAND Memory Device in Byte or 16-Bit Word Stream Mode

NAND devices require correct command and address programming before data array read or write accesses. The GPMC does not include specific hardware to translate a random address system request into a NAND-specific multiphase access. In that sense, GPMC NAND support, as opposed to random memory-map device support, is data-stream-oriented (byte or 16-bit word).

The GPMC NAND programming model relies on a software driver for address and command formatting with the correct data address pointer value according to the block and page structure. Because of NAND structure and protocol interface diversity, the GPMC does not support automatic command and address phase programming, and software drivers must access the NAND device ID to ensure that correct command and address formatting are used for the identified device.

NAND device data read and write accesses are achieved through an asynchronous read or write access. The associated chip-select signal timing control must be programmed according to the NAND device timing specification.

Copyright © 2011, Texas Instruments Incorporated

Any chip-select region can be qualified as a NAND region to constrain the $\overline{\text{ADV}}$_ALE signal as Address Latch Enable (ALE active high, default state value at low) during address program access, and the $\overline{\text{BE0}}$_CLE signal as Command Latch Enable (CLE active high, default state value at low) during command program access. GPMC address lines are not used (the previous value is not changed) during NAND access.

6.1.3.4.12.1.1 Chip-Select Configuration for NAND Interfacing in Byte or Word Stream Mode

The GPMC_CONFIG7_i register (where i = 0 to 7) associated with a NAND device region interfaced in byte or word stream mode can be initialized with a minimum size of 16 Mbytes, because any address location in the chip-select memory region can be used to access a NAND data array. The NAND Flash protocol specifies an address sequence where address bits are passed through the data bus in a series of write accesses with the ALE pin asserted. After this address phase, all operations are streamed and the system requests address is irrelevant.

To allow correct command, address, and data-access controls, the GPMC_CONFIG1_i register associated with a NAND device region must be initialized in asynchronous read and write modes with the parameters shown in Table 6-15. Failure to comply with these settings corrupts the NAND interface protocol.

The GPMC_CONFIG1_i to GPMC_CONFIG4_i register (where i = 0 to 7) associated with a NAND device region must be initialized with the correct control-signal timing value according to the NAND device timing parameters.

Table 6-15. Chip-Select Configuration for NAND Interfacing

Bit Field	Register	Value	Comments
WRAPBURST	GPMC_CONFIG1_i	0	No wrap
READMULTIPLE	GPMC_CONFIG1_i	0	Single access
READTYPE	GPMC_CONFIG1_i	0	Asynchronous mode
WRITEMULTIPLE	GPMC_CONFIG1_i	0	Single access
WRITETYPE	GPMC_CONFIG1_i	0	Asynchronous mode
CLKACTIVATIONTIME	GPMC_CONFIG1_i	0b00	
ATTACHEDDEVICEPAGELENGTH	GPMC_CONFIG1_i	Don't care	Single-access mode
WAITREADMONITORING	GPMC_CONFIG1_i	0	Wait not monitored by GPMC access engine
WAITWRITEMONITORING	GPMC_CONFIG1_i	0	Wait not monitored by GPMC access engine
WAITMONITORINGTIME	GPMC_CONFIG1_i	Don't care	Wait not monitored by GPMC access engine
WAITPINSELECT	GPMC_CONFIG1_i		Select which wait is monitored by edge detectors
DEVICESIZE	GPMC_CONFIG1_i	0b00 or 0b01	8- or 16-bit interface
DEVICETYPE	GPMC_CONFIG1_i	0b10	NAND device in stream mode
MUXADDDATA	GPMC_CONFIG1_i	0b00	Nonmultiplexed mode
TIMEPARAGRANULARITY	GPMC_CONFIG1_i	0	Timing achieved with best GPMC clock granularity
GPMCFCLKDIVIDER	GPMC_CONFIG1_i	Don't care	Asynchronous mode

6.1.3.4.12.1.2 NAND Device Command and Address Phase Control

NAND devices require multiple address programming phases. The MPU software driver is responsible for issuing the correct number of command and address program accesses, according to the device command set and the device address-mapping scheme.

NAND device-command and address-phase programming is achieved through write requests to the GPMC_NAND_COMMAND_i and GPMC_NAND_ADDRESS_i register locations (i = 0 to 7) with the correct command and address values. These locations are mapped in the associated chip-select register region. The associated chip-select signal timing control must be programmed according to the NAND device timing specification.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Command and address values are not latched during the access and cannot be read back at the register location.

- Only write accesses must be issued to these locations, but the GPMC does not discard any read access. Accessing a NAND device with $\overline{OE}$ and CLE or ALE asserted (read access) can produce undefined results.
- Write accesses to the GPMC_NAND_COMMAND_i register location and to the GPMC_NAND_ADDRESS_i register location must be posted for faster operations (i = 0 to 7). The GPMC_CONFIG[0] NANDFORCEPOSTEDWRITE bit enables write accesses to these locations as posted, even if they are defined as nonposted.

A write buffer is used to store write transaction information before the external device is accessed:

- Up to eight consecutive posted write accesses can be accepted and stored in the write buffer.
- For nonposted write, the pipeline is one deep.
- A GPMC_STATUS[0] EMPTYWRITEBUFFERSTATUS bit stores the empty status of the write buffer.

The GPMC_NAND_COMMAND_i and GPMC_NAND_ADDRESS_i registers (where i = 0 to 7) are 32-bit word locations, which means any 32-bit word or 16-bit word access is split into 4- or 2-byte accesses if an 8-bit wide NAND device is attached. For multiple-command phase or multiple-address phase, the software driver can use 32-bit word or 16-bit word access to these registers, but it must account for the splitting and little-endian ordering scheme. When only one byte command or address phase is required, only byte write access to a GPMC_NAND_COMMAND_i and GPMC_NAND_ADDRESS_i can be used, and any of the four byte locations of the registers are valid.

The same applies to GPMC_NAND_COMMAND_i and GPMC_NAND_ADDRESS_i (i = 0 to 7) 32-bit word write access to a 16-bit wide NAND device (split into two 16-bit word accesses). In the case of a 16-bit word write access, the MSByte of the 16-bit word value must be set according to the NAND device requirement (usually 0). Either 16-bit word location or any one of the four byte locations of the registers is valid

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.12.1.3 Command Latch Cycle

Writing data at the GPMC_NAND_COMMAND_i location (i = 0 to 7) places the data as the NAND command value on the bus, using a regular asynchronous write access.

- $\overline{CE}$ is controlled by the CSONTIME and CSWROFFTIME timing parameters.
- CLE is controlled by the ADVONTIME and ADVWROFFTIME timing parameters.
- WE is controlled by the WEONTIME and WEOFFTIME timing parameters.
- ALE and $\overline{RE}$ ($\overline{OE}$) are maintained inactive.

Figure 6-28 shows the NAND command latch cycle.

CLE is shared with the $\overline{BE0}$ output signal and has an inverted polarity from $\overline{BE0}$. The NAND qualifier deals with this. During the asynchronous NAND data access cycle, $\overline{BE0}$ (also $\overline{BE1}$) must not toggle, because it is shared with CLE.

NAND Flash memories do not use byte enable signals at all.

Figure 6-28. NAND Command Latch Cycle

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.12.1.4 Address Latch Cycle

Writing data at the GPMC_NAND_ADDRESS_i location (i = 0 to 7) places the data as the NAND partial address value on the bus, using a regular asynchronous write access.

- $\overline{CS}$ is controlled by the CSONTIME and CSWROFFTIME timing parameters.
- ALE is controlled by the ADVONTIME and ADVWROFFTIME timing parameters.
- $\overline{WE}$ is controlled by the WEONTIME and WEOFFTIME timing parameters.
- CLE and $\overline{RE}$ ($\overline{OE}$) are maintained inactive.

Figure 6-29 shows the NAND address latch cycle.

ALE is shared with the $\overline{ADV}$ output signal and has an inverted polarity from $\overline{ADV}$. The NAND qualifier deals with this. During the asynchronous NAND data access cycle, ALE is kept stable.

Figure 6-29. NAND Address Latch Cycle

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.12.1.5 NAND Device Data Read and Write Phase Control in Stream Mode

NAND device data read and write accesses are achieved through a read or write request to the chip-select-associated memory region at any address location in the region or through a read or write request to the GPMC_NAND_DATA_i location (i = 0 to 7) mapped in the chip-select-associated control register region. GPMC_NAND_DATA_i is not a true register, but an address location to enable $\overline{RE}$ or $\overline{WE}$ signal control. The associated chip-select signal timing control must be programmed according to the NAND device timing specification.

Reading data from the GPMC_NAND_DATA_i location or from any location in the associated chip-select memory region activates an asynchronous read access.

- $\overline{CS}$ is controlled by the CSONTIME and CSRDOFFTIME timing parameters.
- $\overline{RE}$ is controlled by the OEONTIME and OEOFFTIME timing parameters.
- To take advantage of $\overline{RE}$ high-to-data invalid minimum timing value, the RDACCESSTIME can be set so that data are effectively captured after $\overline{RE}$ deassertion. This allows optimization of NAND read access cycle time completion. For optimal timing parameter settings, see the NAND device and the device IC timing parameters.

ALE, CLE, and $\overline{WE}$ are maintained inactive.

Figure 6-30 shows the NAND data read cycle.

Figure 6-30. NAND Data Read Cycle

Writing data to the GPMC_NAND_DATA_i location or to any location in the associated chip-select memory region activates an asynchronous write access.

- $\overline{CS}$ is controlled by the CSONTIME and CSWROFFTIME timing parameters.
- $\overline{WE}$ is controlled by the WEONTIME and WEOFFTIME timing parameters.
- ALE, CLE, and $\overline{RE}$ ($\overline{OE}$) are maintained inactive.

Figure 6-31 shows the NAND data write cycle.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 6-31. NAND Data Write Cycle

6.1.3.4.12.1.6 NAND Device General Chip-Select Timing Control Requirement

For most NAND devices, read data access time is dominated by $\overline{CS}$-to-data-valid timing and has faster $\overline{RE}$-to-data-valid timing. Successive accesses with $\overline{CS}$ deassertions between accesses are affected by this timing constraint. Because accesses to a NAND device can be interleaved with other chip-select accesses, there is no certainty that $\overline{CS}$ always stays low between two accesses to the same chip-select. Moreover, an $\overline{CS}$ deassertion time between the same chip-select NAND accesses is likely to be required as follows: the $\overline{CS}$ deassertion requires programming CYCLETIME and RDACCESSTIME according to the $\overline{CS}$-to-data-valid critical timing.

To get full performance from NAND read and write accesses, the prefetch engine can dynamically reduce RDCYCLETIME, WRCYCLETIME, RDACCESSTIME, WRACCESSTIME, CSRDOFFTIME, CSWROFFTIME, ADVRDOFFTIME, ADVWROFFTIME, OEOFFTIME, and WEOFFTIME on back-to-back NAND accesses (to the same memory) and suppress the minimum $\overline{CS}$ high pulse width between accesses. For more information about optimal prefetch engine access, see Section 6.1.3.4.12.4.

Some NAND devices require minimum write-to-read idle time, especially for device-status read accesses following status-read command programming (write access). If such write-to-read transactions are used, a minimum $\overline{CS}$ high pulse width must be set. For this, CYCLE2CYCLESAMECSEN and CYCLE2CYCLEDELAY must be set according to the appropriate timing requirement to prevent any timing violation.

NAND devices usually have an important $\overline{RE}$ high to data bus in tristate mode. This requires a bus turnaround setting (BUSTURNAROUND = 1), so that the next access to a different chip-select is delayed until the BUSTURNAROUND delay completes. Back-to-back NAND read accesses to the same NAND Flash are not affected by the programmed bus turnaround delay.

6.1.3.4.12.1.7 Read and Write Access Size Adaptation

6.1.3.4.12.1.7.1 8-Bit Wide NAND Device

Host 16-bit word and 32-bit word read and write access requests to a chip-select associated with an 8-bit wide NAND device are split into successive read and write byte accesses to the NAND memory device. Byte access is ordered according to little-endian organization. A NAND 8-bit wide device must be interfaced on the D0D7 interface bus lane. GPMC data accesses are justified on this bus lane when the chip-select is associated with an 8-bit wide NAND device.

6.1.3.4.12.1.7.2 16-Bit Wide NAND Device

Host 32-bit word read and write access requests to a chip-select associated with a 16-bit wide NAND device are split into successive read and write 16-bit word accesses to the NAND memory device. 16-bit word access is ordered according to little-endian organization.

Copyright © 2011, Texas Instruments Incorporated

Host byte read and write access requests to a 16-bit wide NAND device are completed as 16-bit accesses on the device itself, because there is no byte-addressing capability on 16-bit wide NAND devices. This means that the NAND device address pointer is incremented on a 16-bit word basis and not on a byte basis. For a read access, only the requested byte is given back to the host, but the remaining byte is not stored or saved by the GPMC, and the next byte or 16-bit word read access gets the next 16-bit word NAND location. For a write access, the invalid byte part of the 16-bit word is driven to FF, and the next byte or 16-bit word write access programs the next 16-bit word NAND location.

Generally, byte access to a 16-bit wide NAND device should be avoided, especially when ECC calculation is enabled. 8-bit or 16-bit ECC-based computations are corrupted by a byte read to a 16-bit wide NAND device, because the nonrequested byte is considered invalid on a read access (not captured on the external data bus; FF is fed to the ECC engine) and is set to FF on a write access.

Host requests (read/write) issued in the chip-select memory region are translated in successive single or split accesses (read/write) to the attached device. Therefore, incrementing 32-bit burst requests are translated in multiple 32-bit sequential accesses following the access adaptation of the 32-bit to 8- or 16-bit device.

6.1.3.4.12.2 NAND Device-Ready Pin

The NAND memory device provides a ready pin to indicate data availability after a block/page opening and to indicate that data programming is complete. The ready pin can be connected to one of the WAIT GPMC input pins; data read accesses must not be tried when the ready pin is sampled inactive (device is not ready) even if the associated chip-select WAITREADMONITORING bit field is set. The duration of the NAND device busy state after the block/page opening is so long (up to 50 µs) that accesses occurring when the ready pin is sampled inactive can stall GPMC access and eventually cause a system time-out.

If a read access to a NAND flash is done using the wait monitoring mode, the device is blocked during a page opening, and so is the GPMC. If the correct settings are used, other chip-selects can be used while the memory processes the page opening command.

To avoid a time-out caused by a block/page opening delay in NAND flash, disable the wait pin monitoring for read and write accesses (that is, set the GPMC_CONFIG1_i[[21] WAITWRITEMONITORING and GPMC_CONFIG1_i[[22] WAITREADMONITORING bits to 0, where i = 0 to 7), and use one of the following methods instead:

- Use software to poll the WAITnSTATUS bit (n = 0 to 1) of the GPMC_STATUS register.
- Configure an interrupt that is generated on the WAIT signal change (through the GPMC_IRQENABLE [11-8]bits).

Even if the READWAITMONITORING bit is not set, the external memory nR/B pin status is captured in the programmed WAIT bit in the GPMC_STATUS register.

The READWAITMONITORING bit method must be used for other memories than NAND flash, if they require the use of a WAIT signal.

6.1.3.4.12.2.1 Ready Pin Monitored by Software Polling

The ready signal state can be monitored through the GPMC_STATUS WAITxSTATUS bit (x = 0 or 1). The software must monitor the ready pin only when the signal is declared valid. Refer to the NAND device timing parameters to set the correct software temporization to monitor ready only after the invalid window is complete from the last read command written to the NAND device.

6.1.3.4.12.2.2 Ready Pin Monitored by Hardware Interrupt

Each gpmc_wait input pin can generate an interrupt when a wait-to-no-wait transition is detected. Depending on whether the GPMC_CONFIG WAITxPINPOLARITY bits (x = 0 or 1) is active low or active high, the wait-to-no-wait transition is a low-to-high external WAIT signal transition or a high-to-low external WAIT signal transition, respectively.

The wait transition pin detector must be cleared before any transition detection. This is done by writing 1

Copyright © 2011, Texas Instruments Incorporated

to the WAITxEDGEDETECTIONSTATUS bit (x = 0 or 1) of the GPMC_IRQSTATUS register according to the gpmc_wait pin used for the NAND device-ready signal monitoring. To detect a wait-to-no-wait transition, the transition detector requires a wait active time detection of a minimum of two GPMC_FCLK cycles. Software must incorporate precautions to clear the wait transition pin detector before wait (busy) time completes.

A wait-to-no-wait transition detection can issue a GPMC interrupt if the WAITxEDGEDETECTIONENABLE bit in the GPMC_IRQENABLE register is set and if the WAITxEDGEDETECTIONSTATUS bit field in the GPMC_IRQSTATUS register is set.

The WAITMONITORINGTIME field does not affect wait-to-no-wait transition time detection.

It is also possible to poll the WAITxEDGEDETECTIONSTATUS bit field in the GPMC_IRQSTATUS register according to the gpmc_wait pin used for the NAND device ready signal monitoring.

6.1.3.4.12.3 ECC Calculator

The General Purpose Memory Controller includes an Error Code Correction (ECC) calculator circuitry that enables on the fly ECC calculation during data read or data program (that is, write) operations. The page size supported by the ECC calculator in one calculation/context is 512 bytes.

The user can choose from two different algorithms with different error correction capabilities through the GPMC_ECC_CONFIG[16] ECCALGORITHM bit:

- Hamming code for 1-bit error code correction on 8- or 16-bit NAND Flash organized with page size greater than 512 bytes
- BCH (Bose-Chaudhurl-Hocquenghem) code for 4- to 16-bit error correction

The GPMC does not directly handle the error code correction itself. During writes, the GPMC computes parity bits. During reads, the GPMC provides enough information for the processor to correct errors without reading the data buffer all over again.

The Hamming code ECC is based on a 2-dimensional (row and column) bit parity accumulation. This parity accumulation is either accomplished on the programmed number of bytes or 16-bit words read from the memory device, or written to the memory device in stream mode.

Because the ECC engine includes only one accumulation context, it can be allocated to only one chip-select at a time through the GPMC_ECC_CONFIG[3-1] ECCCS bit field. Even if two CS use different ECC algorithms, one the Hamming code and the other a BCH code, they must define separate ECC contexts because some of the ECC registers are common to all types of algorithms.

6.1.3.4.12.3.1 Hamming Code

All references to Error Code Correction (ECC) in this subsection refer to the 1-bit error correction Hamming code.

The ECC is based on a two-dimensional (row and column) bit parity accumulation known as Hamming Code. The parity accumulation is done for a programmed number of bytes or 16-bit word read from the memory device or written to the memory device in stream mode.

There is no automatic error detection or correction, and it is the software NAND driver responsibility to read the multiple ECC calculation results, compare them to the expected code value, and take the appropriate corrective actions according to the error handling strategy (ECC storage in spare byte, error correction on read, block invalidation).

The ECC engine includes a single accumulation context. It can be allocated to a single designated chip-select at a time and parallel computations on different chip-selects are not possible. Since it is allocated to a single chip-select, the ECC computation is not affected by interleaved GPMC accesses to other chip-selects and devices. The ECC accumulation is sequentially processed in the order of data read from or written to the memory on the designated chip-select. The ECC engine does not differentiate read accesses from write accesses and does not differentiate data from command or status information. It is the software responsibility to make sure only relevant data are passed to the NAND flash memory while the ECC computation engine is active.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

The starting NAND page location must be programmed first, followed by an ECC accumulation context reset with an ECC enabling, if required. The NAND device accesses discussed in the following sections must be limited to data read or write until the specified number of ECC calculations is completed.

6.1.3.4.12.3.1.1 ECC Result Register and ECC Computation Accumulation Size

The GPMC includes up to nine ECC result registers (GPMC_ECCj_RESULT, j = 1 to 9) to store ECC computation results when the specified number of bytes or 16-bit words has been computed.

The ECC result registers are used sequentially; one ECC result is stored in one ECC result register on the list, the next ECC result is stored in the next ECC result register on the list, and so forth, until the last ECC computation. The value of the GPMC_ECCj_RESULT register value is valid only when the programmed number of bytes or 16-bit words has been accumulated, which means that the same number of bytes or 16-bit words has been read from or written to the NAND device in sequence.

The GPMC_ECC_CONTROL[3-0] ECCPOINTER field must be set to the correct value to select the ECC result register to be used first in the list for the incoming ECC computation process. The ECCPointer can be read to determine which ECC register is used in the next ECC result storage for the ongoing ECC computation. The value of the GPMC_ECCj_RESULT register (j = 1 to 9) can be considered valid when ECCPOINTER equals j + 1. When the GPMC_ECCj_RESULT (where j = 9) is updated, ECCPOINTER is frozen at 10, and ECC computing is stopped (ECCENABLE = 0).

The ECC accumulator must be reset before any ECC computation accumulation process. The GPMC_ECC_CONTROL[8] ECCCLEAR bit must be set to 1 (nonpersistent bit) to clear the accumulator and all ECC result registers.

For each ECC result (each register, j = 1 to 9), the number of bytes or 16-bit words used for ECC computing accumulation can be selected from between two programmable values.

The ECCjRESULTSIZE bits (j = 1 to 9) in the GPMC_ECC_SIZE_CONFIG register select which programmable size value (ECCSIZE0 or ECCSIZE1) must be used for this ECC result (stored in GPMC_ECCj_RESULT register).

The ECCSIZE0 and ECCSIZE1 fields allow selection of the number of bytes or 16-bit words used for ECC computation accumulation. Any even values from 2 to 512 are allowed.

Flexibility in the number of ECCs computed and the number of bytes or 16-bit words used in the successive ECC computations enables different NAND page error-correction strategies. Usually based on 256 or 512 bytes and on 128 or 256 16-bit word, the number of ECC results required is a function of the NAND device page size. Specific ECC accumulation size can be used when computing the ECC on the NAND spare byte.

For example, with a 2 Kbyte data page 8-bit wide NAND device, eight ECCs accumulated on 256 bytes can be computed and added to one extra ECC computed on the 24 spare bytes area where the eight ECC results used for comparison and correction with the computed data page ECC are stored. The GPMC then provides nine GPMC_ECCj_RESULT registers (j= 1 to 9) to store the results. In this case, ECCSIZE0 is set to 256, and ECCSIZE1 is set to 24; the ECC[1-8]RESULTSIZE bits are cleared to 0, and the ECC9RESULTSIZE bit is set to 1.

6.1.3.4.12.3.1.2 ECC Enabling

The GPMC_ECC_CONFIG[3-1] ECCCS field selects the allocated chip-select. The GPMC_ECC_CONFIG[0] ECCENABLE bit enables ECC computation on the next detected read or write access to the selected chip-select.

The ECCPOINTER, ECCCLEAR, ECCSIZE, ECCjRESULTSIZE (where j = 1 to 9), ECC16B, and ECCCS fields must not be changed or cleared while an ECC computation is in progress.

The ECC accumulator and ECC result register must not be changed or cleared while an ECC computation is in progress.

Table 6-16 describes the ECC enable settings.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 6-16. ECC Enable Settings

Bit Field	Register	Value	Comments
ECCCS	GPMC_ECC_CONFIG	0-3h	Selects the chip-select where ECC is computed
ECC16B	GPMC_ECC_CONFIG	0/1	Selects column number for ECC calculation
ECCCLEAR	GPMC_ECC_CONTROL	0-7h	Clears all ECC result registers
ECCPOINTER	GPMC_ECC_CONTROL	0-7h	A write to this bit field selects the ECC result register where the first ECC computation is stored. Set to 1 by default.
ECCSIZE1	GPMC_ECC_SIZE_CONFIG	0-FFh	Defines ECCSIZE1
ECCSIZE0	GPMC_ECC_SIZE_CONFIG	0-FFh	Defines ECCSIZE0
ECCjRESULTSIZE (j from 1 to 9)	GPMC_ECC_SIZE_CONFIG	0/1	Selects the size of ECCn result register
ECCENABLE	GPMC_ECC_CONFIG	1	Enables the ECC computation

6.1.3.4.12.3.1.3 ECC Computation

The ECC algorithm is a multiple parity bit accumulation computed on the odd and even bit streams extracted from the byte or Word 16 streams. The parity accumulation is split into row and column accumulations, as shown in Figure 6-32 and Figure 6-33. The intermediate row and column parities are used to compute the upper level row and column parities. Only the final computation of each parity bit is used for ECC comparison and correction.

P1o = bit7 XOR bit5 XOR bit3 XOR bit1 on each byte of the data stream

P1e = bit6 XOR bit4 XOR bit2 XOR bit0 on each byte of the data stream

P2o = bit7 XOR bit6 XOR bit3 XOR bit2 on each byte of the data stream

P2e = bit5 XOR bit4 XOR bit1 XOR bit0 on each byte of the data stream

P4o = bit7 XOR bit6 XOR bit5 XOR bit4 on each byte of the data stream

P4e = bit3 XOR bit2 XOR bit1 XOR bit0 on each byte of the data stream

Each column parity bit is XORed with the previous accumulated value.

Figure 6-32. Hamming Code Accumulation Algorithm (1 of 2)

Copyright © 2011, Texas Instruments Incorporated

For line parities, the bits of each new data are XORed together, and line parity bits are computed as:

P8e = row0 XOR row2 XOR row4 XOR ... XOR row254

P8o = row1 XOR row3 XOR row5 XOR ... XOR row255

P16e = row0 XOR row1 XOR row4 XOR row5 XOR ... XOR row252 XOR row 253

P16o = row2 XOR row3 XOR row6 XOR row7 XOR ... XOR row254 XOR row 255

Unused parity bits in the result registers are cleared to 0.

Figure 6-33. Hamming Code Accumulation Algorithm (2 of 2)

Figure 6-34 shows ECC computation for a 256-byte data stream (read or write). The result includes six column parity bits (P1o-P2o-P4o for odd parities, and P1e-P2e-P4e for even parities) and sixteen row parity bits (P8o-P16o-P32o--P1024o for odd parities, and P8e-P16e-P32e--P1024e for even parities).

Figure 6-34. ECC Computation for a 256-Byte Data Stream (Read or Write)

Copyright © 2011, Texas Instruments Incorporated

Figure 6-35 shows ECC computation for a 512-byte data stream (read or write). The result includes six column parity bits (P1o-P2o-P4o for odd parities, and P1e-P2e-P4e for even parities) and eighteen row parity bits (P8o-P16o-P32o--P1024o- - P2048o for odd parities, and P8e-P16e-P32e--P1024e- P2048e for even parities).

For a 2 Kbytes page, four 512 bytes ECC calculations plus one for the spare area are required. Results are stored in the GPMC_ECCj_RESULT registers (j = 1 to 9).

Figure 6-35. ECC Computation for a 512-Byte Data Stream (Read or Write)

6.1.3.4.12.3.1.4 ECC Comparison and Correction

To detect an error, the computed ECC result must be XORed with the parity value stored in the spare area of the accessed page.

- If the result of this logical XOR is all 0s, no error is detected and the read data is correct.
- If every second bit in the parity result is a 1, one bit is corrupted and is located at bit address (P2048o, P1024o, P512o, P256o, P128o, P64o, P32o, P16o, P8o, P4o, P2o, P1o). The software must correct the corresponding bit.
- If only one bit in the parity result is 1, it is an ECC error and the read data is correct.

6.1.3.4.12.3.1.5 ECC Calculation Based on 8-Bit Word

The 8-bit based ECC computation is used for 8-bit wide NAND device interfacing.

The 8-bit based ECC computation can be used for 16-bit wide NAND device interfacing to get backward compatibility on the error-handling strategy used with 8-bit wide NAND devices. In this case, the 16-bit wide data read from or written to the NAND device is fragmented into 2 bytes. According to little-endian access, the least significant bit (LSB) of the 16-bit wide data is ordered first in the byte stream used for 8-bit based ECC computation.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.12.3.1.6 ECC Calculation Based on 16-Bit Word

ECC computation based on a 16-bit word is used for 16-bit wide NAND device interfacing. This ECC computation is not supported when interfacing an 8-bit wide NAND device, and the GPMC_ECC_CONFIG[7] ECC16B bit must be cleared to 0 when interfacing an 8-bit wide NAND device.

The parity computation based on 16-bit words affects the row and column parity mapping. The main difference is that the odd and even parity bits P8o and P8e are computed on rows for an 8-bit based ECC while there are computed on columns for a 16-bit based ECC. Figure 6-36 and Figure 6-37.

Figure 6-36. 128 Word16 ECC Computation

Figure 6-37. 256 Word16 ECC Computation

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.12.3.2 BCH Code (Bose-Chaudhurl-Hocquenghem)

All references to Error Code Correction (ECC) in this subsection refer to the 4- to 16-bit error correction BCH code.

6.1.3.4.12.3.2.1 Requirements

Read and write accesses to a NAND flash take place by whole pages, in a predetermined sequence: first the data byte page itself, then some spare bytes, including the BCH ECC (and other information). The NAND IC can cache a full page, including spares, for read and write accesses.

Typical page write sequence:

- Sequential write to NAND cache of main data + spare data, for a page. ECC is calculated on the fly. Calculated ECC may be inserted on the fly in the spares, or replaced by dummy accesses.
- When the calculated ECC is replaced by dummy accesses, it must be written to the cache in a second, separate phase. The ECC module is disabled during that time.
- NAND writes its cache line (page) to the array

Typical page read sequence:

- Sequential read of a page. ECC is calculated on the fly.
- ECC module buffers status determines the presence of errors.
- Accesses to several memories may be interleaved by the GPMC, but only one of those memories can be a NAND using the BCH engine at a time; in other words, only one BCH calculation (for example, for a single page) can be on-going at any time. Note also that the sequential nature of NAND accesses guarantees that the data is always written / read out in the same order. BCH-relevant accesses are selected by the GPMCs chip-select.
- Each page may hold up to 4 Kbytes of data, spare bytes not included. This means up to 8 x 512-byte BCH messages. Since all the data is written / read out first, followed by the BCH ECC, this means that the BCH engine must be able to hold 8 104-bit remainders or syndromes (or smaller, 52-bit ones) at the same time.

The BCH module has the capacity to store all remainders internally. After the page start, an internal counter is used to detect the 512-byte sector boundaries. On those boundaries, the current remainder is stored and the divider reset for the next calculation. At the end of the page, the BCH module contains all remainders.

- NAND access cycles hold 8 or 16 bits of data each (1 or 2 bytes); Each NAND cycle takes at least 4 cycles of the GPMCs internal clock. This means the NAND flash timing parameters must define a RDCYCLETIME and a WRCYCLETIME of at least 4 clock cycles after optimization when using the BCH calculator.
- The spare area is assumed to be large enough to hold the BCH ECC, that is, to have at least a message of 13 bytes available per 512-byte sector of data. The zone of unused spare area by the ECC may or may not be protected by the same ECC scheme, by extending the BCH message beyond 512 bytes (maximum codeword is 1023-byte long, ECC included, which leaves a lot of space to cover some spares bytes).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.12.3.2.2 Memory-Mapping of the BCH Codeword

BCH encoding considers a block of data to protect as a polynomial message M(x). In our standard case, 512 bytes of data (that is, 2 bits = 4096 bits) are seen as a polynomial of degree 2 - 1 = 4095, with parameters ranging from M0 to M4095. For 512 bytes of data, 52 bits are required for 4-bit error correction, and 104 bits are required for 8-bit error correction and 207 bits are required for 16-bit error correction. The ECC is a remainder polynomial R(x) of degree 103 (or 51, depending on the selected mode). The complete codeword C(x) is the concatenation of M(x) and R(x) as shown in Table 6-17.

Table 6-17. Flattened BCH Codeword Mapping (512 Bytes + 104 Bits)

	Message M(x)	ECC R(x)				
Bit number	M4095	...	M0	R103	...	R0

If the message is extended by the addition of spare bytes to be protected by the same ECC, the principle is still valid. For example, a 3-byte extension of the message gives a polynomial message M(x) of degree ((512 + 3) × 8) - 1 = 4119, for a total of 3 + 13 = 16 spare bytes of spare, all protected as part of the same codeword.

The message and the ECC bits are manipulated and mapped in the GPMC byte-oriented system. The ECC bits are stored in , , , and (where i = 0 to 7).

- GPMC_BCH_RESULT0_i
- GPMC_BCH_RESULT1_i
- GPMC_BCH_RESULT2_i
- GPMC_BCH_RESULT3_i

6.1.3.4.12.3.2.3 Memory Mapping of the Data Message

The data message mapping shall follow the following rules:

- Bit endianness within a byte is little-endian, that is, the bytes LS bit is also the lowest-degree polynomial parameter: a byte b7-b0 (with b0 the LS bit) represents a segment of polynomial b7 * x + b6 * x + ... + b0 * x
- The message is mapped in the NAND starting with the highest-order parameters, that is, in the lowest addresses of a NAND page.
- Byte endianness within the NANDs 16-bit words is big endian. This means that the same message mapped in 8- and 16-bit memories has the same content at the same byte address.

The BCH module has no visibility over actual addresses. The most important point is the sequence of data word the BCH sees. However, the NAND page is always scanned incrementally in read and write accesses, and this produces the mapping patterns described in the following.

Table 6-18 and Table 6-19 show the mapping of the same 512-byte vector (typically a BCH message) in the NAND memory space. Note that the byte 'address' is only an offset modulo 512 (200h), since the same page may contain several contiguous 512-byte sectors (BCH blocks). The LSB and MSB are respectively the bits M0 and M(2^12-1) of the codeword mapping given above. In both cases the data vectors are aligned, that is, their boundaries coincide with the RAMs data word boundaries.

Table 6-18. Aligned Message Byte Mapping in 8-bit NAND

Byte Offset	8-Bit Word
0	(msb) Byte 511 (1FFh)
1h	Byte 510 (1FEh)
⋮	⋮
1FFh	Byte 0 (0) (LSB)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 6-19. Aligned Message Byte Mapping in 16-bit NAND

Byte Offset	16-Bit Words MSB	16-Bit Words LSB
0	Byte 510 (1FEh)	(msb) Byte 511 (1FFh)
2h	Byte 508 (1FCh)	Byte 509 (1FDh)
⋮	⋮	⋮
1FEh	Byte 0 (0)	(lsb) Byte 1 (1)

Table 6-20 and Table 6-21 show the mapping in memory of arbitrarily-sized messages, starting on access (byte or 16-bit word) boundaries for more clarity. Note that message may actually start and stop on arbitrary nibbles. A nibble is a 4-bit entity. The unused nibbles are not discarded, and they can still be used by the BCH module, but as part of the next message section (for example, on another sectors ECC).

Table 6-20. Aligned Nibble Mapping of Message in 8-bit NAND

Byte Offset	8-Bit Word	
	4-Bit Most Significant Nibble	4-Bit Less Significant Nibble
1	(MSB) Nibble S-1	Nibble S-2
2	Nibble S-3	Nibble S-4
⋮	⋮	⋮
S/2 - 2	Nibble 3	Nibble 2
S/2 - 1	Nibble 1	Nibble 0 (LSB)

Table 6-21. Misaligned Nibble Mapping of Message in 8-bit NAND

Byte Offset	8-Bit Word	
	4-Bit Most Significant Nibble	4-Bit Less Significant Nibble
1	(MSB) Nibble S-1	Nibble S-2
2	Nibble S-3	Nibble S-4
⋮	⋮	⋮
(S+1)/2 - 2	Nibble 2	Nibble 1
(S+1)/2 - 1	Nibble 0 (LSB)	

Table 6-22. Aligned Nibble Mapping of Message in 16-bit NAND

Byte Offset	16-Bit Word			
	4-Bit Most Significant Nibble		4-Bit Less Significant Nibble	
0	Nibble S-3	Nibble S-4	(MSB) Nibble S-1	Nibble S-2
2	Nibble S-7	Nibble S-8	Nibble S-5	Nibble S-6
⋮	⋮	⋮	⋮	⋮
S/2 - 4	Nibble 5	Nibble 4	Nibble 7	Nibble 6
S/2 - 2	Nibble 1	Nibble 0 (LSB)	Nibble 3	Nibble 2

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 6-23. Misaligned Nibble Mapping of Message in 16-bit NAND (1 Unused Nibble)

Byte Offset	16-Bit Word			
	4-Bit Most Significant Nibble		4-Bit Less Significant Nibble	
0	Nibble S-3	Nibble S-4	(MSB) Nibble S-1	Nibble S-2
2	Nibble S-7	Nibble S-8	Nibble S-5	Nibble S-6
⋮	⋮	⋮	⋮	⋮
(S+1)/2 - 4	Nibble 4	Nibble 3	Nibble 6	Nibble 5
(S+1)/2 - 2	Nibble 0 (LSB)		Nibble 2	Nibble 1

Table 6-24. Misaligned Nibble Mapping of Message in 16-bit NAND (2 Unused Nibble)

Byte Offset	16-Bit Word			
	4-Bit Most Significant Nibble		4-Bit Less Significant Nibble	
0	Nibble S-3	Nibble S-4	(MSB) Nibble S-1	Nibble S-2
2	Nibble S-7	Nibble S-8	Nibble S-5	Nibble S-6
⋮	⋮	⋮	⋮	⋮
(S+2)/2 - 4	Nibble 3	Nibble 2	Nibble 5	Nibble 4
(S+2)/2 - 2			Nibble 1	Nibble 0 (LSB)

Table 6-25. Misaligned Nibble Mapping of Message in 16-bit NAND (3 Unused Nibble)

Byte Offset	16-Bit Word			
	4-Bit Most Significant Nibble		4-Bit Less Significant Nibble	
0	Nibble S-3	Nibble S-4	(MSB) Nibble S-1	Nibble S-2
2	Nibble S-7	Nibble S-8	Nibble S-5	Nibble S-6
⋮	⋮	⋮	⋮	⋮
(S+3)/2 - 4	Nibble 2	Nibble 1	Nibble 4	Nibble 3
(S+3)/2 - 2			Nibble 0 (LSB)	

Note that many other cases exist than the ones represented above, for example, where the message does not start on a word boundary.

6.1.3.4.12.3.2.4 Memory Mapping of the ECC

The ECC (or remainder) is presented by the BCH module as a single 104-bit (or 52-bit), little-endian vector. It is up to the software to fetch those 13 bytes (or 6 bytes) from the modules interface, then store them to the NANDs spare area (page write) or to an intermediate buffer for comparison with the stored ECC (page read). There are no constraints on the ECC mapping inside the spare area: it is a softwarecontrolled operation.

However, it is advised to maintain a coherence in the respective formats of the message or the ECC remainder once they have been read out of the NAND. The error correction algorithm works from the complete codeword (concatenated message and remainder) once an error as been detected. The creation of this codeword should be made as straightforward as possible.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

There are cases where the same NAND access contains both data and the ECC protecting that data. This is the case when the data/ECC boundary (which can be on any nibble) does not coincide with an access boundary. The ECC is calculated on-the-fly following the write. In that case, the write must also contain part of the ECC because it is impossible to insert the ECC on-the-fly. Instead:

- During the initial page write (BCH encoding), the ECC is replaced by dummy bits. The BCH encoder is by definition turned OFF during the ECC section, so the BCH result is unmodified.
- During a second phase, the ECC is written to the correct location, next to the actual data.
- The completed line buffer is then written to the NAND array.

6.1.3.4.12.3.2.5 Wrapping Modes

For a given wrapping mode, the module automatically goes through a specific number of sections, as data is being fed into the module. For each section, the BCH core can be enabled (in which case the data is fed to the BCH divider) or not (in which case the BCH simply counts to the end of the section). When enabled, the data is added to the ongoing calculation for a given sector number (for example, number 0).

Wrapping modes are described below. To get a better understanding and see the real-life read and write sequences implemented with each mode, see Section 6.1.3.4.12.3.3.

For each mode:

- A sequence describes the mode in pseudo-language, with for each section the size and the buffer used for ECC processing (if ON). The programmable lengths are size, size0 and size1.
- A checksum condition is given. If the checksum condition is not respected for a given mode, the modules behavior is unpredictable. S is the number of sectors in the page; size0 and size1 are the section sizes programmed for the mode, in nibbles.

Note that wrapping modes 8, 9, 10, and 11 insert a 1-nibble padding where the BCH processing is OFF. This is intended for t = 4 ECC, where ECC is 6 bytes long and the ECC area is expected to include (at least) 1 unused nibble to remain byte-aligned.

6.1.3.4.12.3.2.6 Manual Mode (0x0)

This mode is intended for short sequences, added manually to a given buffer through the software data port input. A complete page may be built out of several such sequences.

To process an arbitrary sequence of 4-bit nibbles, accesses to the software data port shall be made, containing the appropriate data. If the sequence end does not coincide with an access boundary (for example, to process 5 nibbles = 20 bits in 16-bit access mode) and those nibbles need to be skipped, a number of unused nibbles shall be programmed in size1 (in the same example: 5 nibbles to process + 3 to discard = 8 nibbles = exactly 2 x 16-bit accesses: we must program size0 = 5, size1 = 3).

Figure 6-38 shows the manual mode sequence and mapping. In this figure, size and size0 are the same parameter.

Figure 6-38. Manual Mode Sequence and Mapping

Copyright © 2011, Texas Instruments Incorporated

Section processing sequence:

- One time with buffer
  - size0 nibbles of data, processing ON
  - size1 nibbles of unused data, processing OFF

Checksum: size0 + size1 nibbles must fit in a whole number of accesses.

In the following sections, S is the number of sectors in the page.

6.1.3.4.12.3.2.7 Mode 0x1

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing ON
  - size1 nibbles spare, processing OFF

Checksum: Spare area size (nibbles) = S - (size0 + size1)

6.1.3.4.12.3.2.8 Mode 0xA (10)

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing ON
  - 1 nibble pad spare, processing OFF
  - size1 nibbles spare, processing OFF

Checksum: Spare area size (nibbles) = S - (size0 + 1 + size1)

6.1.3.4.12.3.2.9 Mode 0x2

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing OFF
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = S - (size0 + size1)

6.1.3.4.12.3.2.10 Mode 0x3

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- One time with buffer 0
  - size0 nibbles spare, processing ON
- Repeat with buffer 0 to S-1
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = size0 + (S - size1)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.12.3.2.11 Mode 0x7

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- One time with buffer 0
  - size0 nibbles spare, processing ON
- Repeat S times (no buffer used)
  - size1 nibbles spare, processing OFF

Checksum: Spare area size (nibbles) = size0 + (S - size1)

6.1.3.4.12.3.2.12 Mode 0x8

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- One time with buffer 0
  - size0 nibbles spare, processing ON
- Repeat with buffer 0 to S-1
  - 1 nibble padding spare, processing OFF
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = size0 + (S - (1+size1))

6.1.3.4.12.3.2.13 Mode 0x4

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- One time (no buffer used)
  - size0 nibbles spare, processing OFF
- Repeat with buffer 0 to S-1
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = size0 + (S - size1)

6.1.3.4.12.3.2.14 Mode 0x9

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- One time (no buffer used)
  - size0 nibbles spare, processing OFF
- Repeat with buffer 0 to S-1
  - 1 nibble padding spare, processing OFF
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = size0 + (S - (1+size1))

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.12.3.2.15 Mode 0x5

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing ON
- Repeat with buffer 0 to S-1
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = S - (size0 + size1)

6.1.3.4.12.3.2.16 Mode 0xB (11)

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing ON
- Repeat with buffer 0 to S-1
  - 1 nibble padding spare, processing OFF
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = S - (size0 + 1 + size1)

6.1.3.4.12.3.2.17 Mode 0x6

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing ON
- Repeat S times (no buffer used)
  - size1 nibbles spare, processing OFF

Checksum: Spare area size (nibbles) = S - (size0 + size1)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.12.3.3 Supported NAND Page Mappings and ECC Schemes

The following rules apply throughout the entire mapping description:

- Main data area (sectors) size is hardcoded to 512 bytes.
- Spare area size is programmable.
- All page sections (of main area data bytes, protected spare bytes, unprotected spare bytes, and ECC) are defined as explained in Section 6.1.3.4.12.3.2.3.

Each one of the following sections shows a NAND page mapping example (per-sector spare mappings, pooled spare mapping, per-sector spare mapping, with ECC separated at the end of the page).

In the mapping diagrams, sections that belong to the same BCH codeword have the same color (blue or green); unprotected sections are not covered (orange) by the BCH scheme.

Below each mapping diagram, a write (encoding) and read (decoding: syndrome generation) sequence is given, with the number of the active buffers at each point in time (yellow). In the inactive zones (grey), no computing is taking place but the data counter is still active.

In Figure 6-39 to Figure 6-41, tables on the left summarize the mode, size0, size1 parameters to program for respectively write and read processing of a page, with the given mapping, where:

- P is the size of spare byte section Protected by the ECC (in nibbles)
- U is the size of spare byte section Unprotected by the ECC (in nibbles)
- E is the size of the ECC itself (in nibbles)
- S is the number of Sectors per page (2 in the current diagrams)

Each time the processing of a BCH block is complete (ECC calculation for write/encoding, syndrome generation for read/decoding, indicated by red arrows), the update pointer is pulsed. Note that the processing for block 0 can be the first or the last to complete, depending on the NAND page mapping and operation (read or write). All examples show a page size of 1kByte + spares, that is, S = 2 sectors of 512 bytes. The same principles can be extended to larger pages by adding more sectors.

The actual BCH codeword size is used during the error location work to restrict the search range: by definition, errors can only happen in the codeword that was actually written to the NAND, and not in the mathematical codeword of n = 2 - 1 = 8191 bits. That codeword (higher-order bits) is all-zero and implicit during computations.

The actual BCH codeword size depends on the mode, on the programmed sizes and on the sector number (all sizes in nibbles):

- Spares mapped and protected per sector (Figure 6-39: see M1-M2-M3-M9-M10):
  - all sectors: (512) + P + E
- Spares pooled and protected by sector 0 (Figure 6-39: see M5-M6):
  - sector 0 codeword: (512) + P + E
  - other sectors: (512) + E
- Unprotected spares (Figure 6-39: see M4-M7-M8-M11-M12):
  - all codewords (512) + E

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.12.3.3.1 Per-Sector Spare Mappings

In these schemes (Figure 6-39), each 512-byte sector of the main area has its own dedicated section of the spare area. The spare area of each sector is composed of:

- ECC, which must be located after the data it protects
- other data, which may or may not be protected by the sectors ECC

Figure 6-39. NAND Page Mapping and ECC: Per-Sector Schemes

PRELIMINARY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.12.3.3.2 Pooled Spare Mapping

In these schemes (Figure 6-40), the spare area is pooled for the page.

- The ECC of each sector is aligned at the end of the spare area.
- The non-ECC spare data may or may not be covered by the ECC of sector 0

Figure 6-40. NAND Page Mapping and ECC: Pooled Spare Schemes

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.12.3.3.3 Per-Sector Spare Mapping, With ECC Separated at the End of the Page

In these schemes (Figure 6-41), each 512-byte sector of the main area is associated with two sections of the spare area.

- ECC section, all aligned at the end of the page
- other data section, aligned before the ECCs, each of which may or may not be protected by its sectors ECC

Figure 6-41. NAND Page Mapping and ECC: Per-Sector Schemes, with Separate ECC

PRELIMINARY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.12.4 Prefetch and Write-Posting Engine

NAND device data access cycles are usually much slower than the MPU system frequency; such NAND read or write accesses issued by the processor will impact the overall system performance, especially considering long read or write sequences required for NAND page loading or programming. To minimize this effect on system performance, the GPMC includes a prefetch and write-posting engine, which can be used to read from or write to any chip-select location in a buffered manner.

The prefetch and write-posting engine is a simplified embedded-access requester that presents requests to the access engine on a user-defined chip-select target. The access engine interleaves these requests with any request coming from the L3 interface; as a default the prefetch and write-posting engine has the lowest priority.

The prefetch and write-posting engine is dedicated to data-stream access (as opposed to random data access); thus, it is primarily dedicated to NAND support. The engine does not include an address generator; the request is limited to chip-select target identification. It includes a 64-byte FIFO associated with a DMA request synchronization line, for optimal DMA-based use.

The prefetch and write-posting engine uses an embedded 64 bytes (32 16-bit word) FIFO to prefetch data from the NAND device in read mode (prefetch mode) or to store host data to be programmed into the NAND device in write mode (write-posting mode). The FIFO draining and filling (read and write) can be controlled either by the MPU through interrupt synchronization (an interrupt is triggered whenever a programmable threshold is reached) or the sDMA through DMA request synchronization, with a programmable request byte size in both prefetch or posting mode.

The prefetch and write-posting engine includes a single memory pool. Therefore, only one mode, read or write, can be used at any given time. In other words, the prefetch and write-posting engine is a single-context engine that can be allocated to only one chip-select at a time for a read prefetch or a write-posting process.

The engine does not support atomic command and address phase programming and is limited to linear memory read or write access. In consequence, it is limited to NAND data-stream access. The engine relies on the MPU NAND software driver to control block and page opening with the correct data address pointer initialization, before the engine can read from or write to the NAND memory device.

Once started, the engine data reads and writes sequencing is solely based on FIFO location availability and until the total programmed number of bytes is read or written.

Any host-concurrent accesses to a different chip-select are correctly interleaved with ongoing engine accesses. The engine has the lowest priority access so that host accesses to a different chip-select do not suffer a large latency.

A round-robin arbitration scheme can be enabled to ensure minimum bandwidth to the prefetch and write-posting engine in the case of back-to-back direct memory requests to a different chip-select. If the GPMC_PREFETCH_CONFIG1[23] PFPWENROUNDROBIN bit is enabled, the arbitration grants the prefetch and write posting engine access to the GPMC bus for a number of requests programmed in the GPMC_PREFETCH_CONFIG1[19-16] PFPWWEIGHTEDPRIO field.

The prefetch/write-posting engine read or write request is routed to the access engine with the chip-select destination ID. After the required arbitration phase, the access engine processes the request as a single access with the data access size equal to the device size specified in the corresponding chip-select configuration.

The destination chip-select configuration must be set to the NAND protocol-compatible configuration for which address lines are not used (the address bus is not changed from its current value). Selecting a different chip-select configuration can produce undefined behavior.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.12.4.1 General Facts About the Engine Configuration

The engine can be configured only if the GPMC_PREFETCH_CONTROL[0] STARTENGINE bit is de-asserted.

The engine must be correctly configured in prefetch or write-posting mode and must be linked to a NAND chip-select before it can be started. The chip-select is linked using the GPMC_PREFETCH_CONFIG1[26-24] ENGINECSSELECTOR field.

In both prefetch and write-posting modes, the engine respectivelly uses byte or 16-bit word access requests for an 8- or 16-bit wide NAND device attached to the linked chip-select. The FIFOTHRESHOLD and TRANSFERCOUNT fields must be programmed accordingly as a number of bytes or a number of 16-bit word.

When the GPMC_PREFETCH_CONFIG1[7] ENABLEENGINE bit is set, the FIFO entry on the L3 interconnect port side is accessible at any address in the associated chip-select memory region. When the ENABLEENGINE bit is set, any host access to this chip-select is rerouted to the FIFO input. Directly accessing the NAND device linked to this chip-select from the host is still possible through these registers (where i = 0 to 7):

- GPMC_NAND_COMMAND_i
- GPMC_NAND_ADDRESS_i
- GPMC_NAND_DATA_i

The FIFO entry on the L3 interconnect port can be accessed with Byte, 16-bit word, or 32-bit word access size, according to little-endian format, even though the FIFO input is 32-bit wide.

The FIFO control is made easier through the use of interrupts or DMA requests associated with the FIFOTHRESHOLD bit field. The GPMC_PREFETCH_STATUS[30-24] FIFOPOINTER field monitors the number of available bytes to be read in prefetch mode or the number of free empty slots which can be written in write-posting mode. The GPMC_PREFETCH_STATUS[13-0] COUNTVALUE field monitors the number of remaining bytes to be read or written by the engine according to the TRANSFERCOUNT value. The FIFOPOINTER and COUNTVALUE bit fields are always expressed as a number of bytes even if a 16-bit wide NAND device is attached to the linked chip-select.

In prefetch mode, when the FIFOPOINTER equals 0, that is, the FIFO is empty, a host read access receives the byte last read from the FIFO as its response. In case of 32-bit word or 16-bit word read accesses, the last byte read from the FIFO is copied the required number of times to fit the requested word size. In write-posting mode, when the FIFOPOINTER equals 0, that is, the FIFO is full, a host write overwrites the last FIFO byte location. There is no underflow or overflow error reporting in the GPMC.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.12.4.2 Prefetch Mode

The prefetch mode is selected when the GPMC_PREFETCH_CONFIG1[0] ACCESSMODE bit is cleared.

The MPU NAND software driver must issue the block and page opening (READ) command with the correct data address pointer initialization before the engine can be started to read from the NAND memory device. The engine is started by asserting the GPMC_PREFETCH_CONTROL[0] STARTENGINE bit. The STARTENGINE bit automatically clears when the prefetch process completes.

If required, the ECC calculator engine must be initialized (i.e., reset, configured, and enabled) before the prefetch engine is started, so that the ECC is correctly computed on all data read by the prefetch engine.

When the GPMC_PREFETCH_CONFIG1[3] SYNCHROMODE bit is cleared, the prefetch engine starts requesting data as soon as the STARTENGINE bit is set. If using this configuration, the host must monitor the NAND device-ready pin so that it only sets the STARTENGINE bit when the NAND device is in a ready state, meaning data is valid for prefetching.

When the SYNCHROMODE bit is set, the prefetch engine starts requesting data when an active to inactive wait signal transition is detected. The transition detector must be cleared before any transition detection; see Section 6.1.3.4.12.2.2. The GPMC_PREFETCH_CONFIG1[5-4] WAITPINSELECTOR field selects which gpmc_wait pin edge detector triggers the prefetch engine in this synchronized mode.

If the STARTENGINE bit is set after the NAND address phase (page opening command), the engine is effectively started only after the actual NAND address phase completion. To prevent GPMC stall during this NAND address phase, set the STARTENGINE bit field before NAND address phase completion when in synchronized mode. The prefetch engine will start when an active to inactive wait signal transition is detected. The STARTENGINE bit is automatically cleared on prefetch process completion.

The prefetch engine issues a read request to fill the FIFO with the amount of data specified by GPMC_PREFETCH_CONFIG2[13-0] TRANSFERCOUNT field.

Table 6-26 describes the prefetch mode configuration.

Table 6-26. Prefetch Mode Configuration

Bit Field	Register	Value	Comments
STARTENGINE	GPMC_PREFETCH_CONTROL	0	Prefetch engine can be configured only if STARTENGINE is cleared to 0.
ENGINECSSELECTOR	GPMC_PREFETCH_CONFIG1	0 to 3h	Selects the chip-select associated with a NAND device where the prefetch engine is active.
ACCESSMODE	GPMC_PREFETCH_CONFIG1	0	Selects prefetch mode
FIFOTHRESHOLD	GPMC_PREFETCH_CONFIG1		Selects the maximum number of bytes read or written by the host on DMA or interrupt request
TRANSFERCOUNT	GPMC_PREFETCH_CONFIG1		Selects the number of bytes to be read or written by the engine to the selected chip-select
SYNCHROMODE	GPMC_PREFETCH_CONFIG1	0/1	Selects when the engine starts the access to the chip-select
WAITPINSELECT	GPMC_PREFETCH_CONFIG1	0 to 1	Selects wait pin edge detector (if GPMC_PREFETCH_CONFIG1[3] SYNCHROMODE = 1)
ENABLEOPTIMIZEDACCESS	GPMC_PREFETCH_CONFIG1	0/1	See Section 6.1.3.4.12.4.6
CYCLEOPTIMIZATION	GPMC_PREFETCH_CONFIG1		Number of clock cycle removed to timing parameters
ENABLEENGINE	GPMC_PREFETCH_CONFIG1	1	Engine enabled
STARTENGINE	GPMC_PREFETCH_CONFIG1	1	Starts the prefetch engine

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.12.4.3 FIFO Control in Prefetch Mode

The FIFO can be drained directly by the MPU or by an eDMA channel.

In MPU draining mode, the FIFO status can be monitored through the GPMC_PREFETCH_STATUS[30-24] FIFOPOINTER field or through the GPMC_PREFETCH_STATUS[16] FIFOTHRESHOLDSTATUS bit. The FIFOPOINTER indicates the current number of available data to be read; FIFOTHRESHOLDSTATUS set to 1 indicates that at least FIFOTHRESHOLD bytes are available from the FIFO.

An interrupt can be triggered by the GPMC if the GPMC_IRQENABLE[0] FIFOEVENTENABLE bit is set. The FIFO interrupt event is logged, and the GPMC_IRQSTATUS[0] FIFOEVENTSTATUS bit is set. To clear the interrupt, the MPU must read all the available bytes, or at least enough bytes to get below the programmed FIFO threshold, and the FIFOEVENTSTATUS bit must be cleared to enable further interrupt events. The FIFOEVENTSTATUS bit must always be reset prior to asserting the FIFOEVENTENABLE bit to clear any out-of-date logged interrupt event. This interrupt generation must be enabled after enabling the STARTENGINE bit.

Prefetch completion can be monitored through the GPMC_PREFETCH_STATUS[13-0] COUNTVALUE field. COUNTVALUE indicates the number of currently remaining data to be requested according to the TRANSFERCOUNT value. An interrupt can be triggered by the GPMC when the prefetch process is complete (that is, COUNTVALUE equals 0) if the GPMC_IRQENABLE[1] TERMINALCOUNTEVENTENABLE bit is set. At prefetch completion, the TERMINALCOUNT interrupt event is also logged, and the GPMC_IRQSTATUS[1] TERMINALCOUNTSTATUS bit is set. To clear the interrupt, the MPU must clear the TERMINALCOUNTSTATUS bit. The TERMINALCOUNTSTATUS bit must always be cleared prior to asserting the TERMINALCOUNTEVENTENABLE bit to clear any out-of-date logged interrupt event.

NOTE: The COUNTVALUE value is only valid when the prefetch engine is active (started), and an interrupt is only triggered when COUNTVALUE reaches 0, that is, when the prefetch engine automatically goes from an active to an inactive state.

The number of bytes to be prefetched (programmed in TRANSFERCOUNT) must be a multiple of the programmed FIFOTHRESHOLD to trigger the correct number of interrupts allowing a deterministic and transparent FIFO control. If this guideline is respected, the number of ISR accesses is always required and the FIFO is always empty after the last interrupt is trigerred. In other cases, the TERMINALCOUNT interrupt must be used to read the remaining bytes in the FIFO (the number of remaining bytes being lower than the FIFOTHRESHOLD value).

In DMA draining mode, the GPMC_PREFETCH_CONFIG1[2] DMAMODE bit must be set so that the GPMC issues a DMA hardware request when at least FIFOTHRESHOLD bytes are ready to be read from the FIFO. The DMA channel owning this DMA request must be programmed so that the number of bytes programmed in FIFOTHRESHOLD is read from the FIFO during the DMA request process. The DMA request is kept active until this number of bytes has effectively been read from the FIFO, and no other DMA request can be issued until the ongoing active request is complete.

In prefetch mode, the TERMINALCOUNT event is also a source of DMA requests if the number of bytes to be prefetched is not a multiple of FIFOTHRESHOLD, the remaining bytes in the FIFO can be read by the DMA channel using the last DMA request. This assumes that the number of remaining bytes to be read is known and controlled through the DMA channel programming model.

Any potentially active DMA request is cleared when the prefetch engine goes from inactive to active prefetch (the STARTENGINE bit is set to 1). The associated DMA channel must always be enabled by the MPU after setting the STARTENGINE bit so that the out-of-date active DMA request does not trigger spurious DMA transfers.

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.12.4.4 Write-Posting Mode

The write-posting mode is selected when the GPMC_PREFETCH_CONFIG1[0] ACCESSMODE bit is set.

The MPU NAND software driver must issue the correct address pointer initialization command (page program) before the engine can start writing data into the NAND memory device. The engine starts when the GPMC_PREFETCH_CONTROL[0] STARTENGINE bit is set to 1. The STARTENGINE bit clears automatically when posting completes. When all data have been written to the NAND memory device, the MPU NAND software driver must issue the second cycle program command and monitor the status for programming process completion (adding ECC handling, if required).

If used, the ECC calculator engine must be started (configured, reset, and enabled) before the posting engine is started so that the ECC parities are properly calculated on all data written by the prefetch engine to the associated chip-select.

In write-posting mode, the GPMC_PREFETCH_CONFIG1[3] SYNCHROMODE bit must be cleared so that posting starts as soon as the STARTENGINE bit is set and the FIFO is not empty.

If the STARTENGINE bit is set after the NAND address phase (page program command), the STARTENGINE setting is effective only after the actual NAND command completion. To prevent GPMC stall during this NAND command phase, set the STARTENGINE bit field before the NAND address completion and ensure that the associated DMA channel is enabled after the NAND address phase.

The posting engine issues a write request when valid data are available from the FIFO and until the programmed GPMC_PREFETCH_CONFIG2[13-0] TRANSFERCOUNT accesses have been completed.

The STARTENGINE bit clears automatically when posting completes. When all data have been written to the NAND memory device, the MPU NAND software driver must issue the second cycle program command and monitor the status for programming process completion. The closing program command phase must only be issued when the full NAND page has been written into the NAND flash write buffer, including the spare area data and the ECC parities, if used.

Table 6-27. Write-Posting Mode Configuration

Bit Field	Register	Value	Comments
STARTENGINE	GPMC_PREFETCH_CONTROL	0	Write-posting engine can be configured only if STARTENGINE is cleared to 0.
ENGINECSSELECTOR	GPMC_PREFETCH_CONFIG1	0 to 3h	Selects the chip-select associated with a NAND device where the prefetch engine is active
ACCESSMODE	GPMC_PREFETCH_CONFIG1	1	Selects write-posting mode
FIFOTHRESHOLD	GPMC_PREFETCH_CONFIG1		Selects the maximum number of bytes read or written by the host on DMA or interrupt request
TRANSFERCOUNT	GPMC_PREFETCH_CONFIG2		Selects the number of bytes to be read or written by the engine from/to the selected chip-select
SYNCHROMODE	GPMC_PREFETCH_CONFIG1	0	Engine starts the access to chip-select as soon as STARTENGINE is set.
ENABLEOPTIMIZEDACCESS	GPMC_PREFETCH_CONFIG1	0/1	See Section 6.1.3.4.12.4.6
CYCLEOPTIMIZATION	GPMC_PREFETCH_CONFIG		
ENABLEENGINE	GPMC_PREFETCH_CONFIG1	1	Engine enabled
STARTENGINE	GPMC_PREFETCH_CONTROL	1	Starts the prefetch engine

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.12.4.5 FIFO Control in Write-Posting Mode

The FIFO can be filled directly by the MPU or by an sDMA channel.

In MPU filling mode, the FIFO status can be monitored through the FIFOPOINTER or through the GPMC_PREFETCH_STATUS[16] FIFOTHRESHOLDSTATUS bit. FIFOPOINTER indicates the current number of available free byte places in the FIFO, and the FIFOTHRESHOLDSTATUS bit, when set, indicates that at least FIFOTHRESHOLD free byte places are available in the FIFO.

An interrupt can be issued by the GPMC if the GPMC_IRQENABLE[0] FIFOEVENTENABLE bit is set. When the interrupt is fired, the GPMC_IRQSTATUS[0] FIFOEVENTSTATUS bit is set. To clear the interrupt, the MPU must write enough bytes to fill the FIFO, or enough bytes to get below the programmed threshold, and the FIFOEVENTSTATUS bit must be cleared to get further interrupt events. The FIFOEVENTSTATUS bit must always be cleared prior to asserting the FIFOEVENTENABLE bit to clear any out-of-date logged interrupt event. This interrupt must be enabled after enabling the STARTENGINE bit

The posting completion can be monitored through the GPMC_PREFETCH_STATUS[13-0] COUNTVALUE field. COUNTVALUE indicates the current number of remaining data to be written based on the TRANSFERCOUNT value. An interrupt is issued by the GPMC when the write-posting process completes (that is, COUNTVALUE equal to 0) if the GPMC_IRQENABLE[1] TERMINALCOUNTEVENTENABLE bit is set. When the interrupt is fired, the GPMC_IRQSTATUS[1] TERMINALCOUNTSTATUS bit is set. To clear the interrupt, the MPU must clear the TERMINALCOUNTSTATUS bit. The TERMINALCOUNTSTATUS bit must always be cleared prior to asserting the TERMINALCOUNTEVENTENABLE bit to clear any out-of-date logged interrupt event.

NOTE: The COUNTVALUE value is only valid if the write-posting engine is active and started, and an interrupt is only issued when COUNTVALUE reaches 0, that is, when the posting engine automatically goes from active to inactive.

In DMA filling mode, the DMAMode bit field in the GPMC_PREFETCH_CONFIG1[2] DMAMODE bit must be set so that the GPMC issues a DMA hardware request when at least FIFOTHRESHOLD bytes-free places are available in the FIFO. The DMA channel owning this DMA request must be programmed so that a number of bytes equal to the value programmed in the FIFOTHRESHOLD bit field are written into the FIFO during the DMA access. The DMA request remains active until the associated number of bytes has effectively been written into the FIFO, and no other DMA request can be issued until the ongoing active request has been completed.

Any potentially active DMA request is cleared when the prefetch engine goes from inactive to active prefetch (STARTENGINE set to 1). The associated DMA channel must always be enabled by the MPU after setting the STARTENGINE bit so that an out-of-date active DMA request does not trigger spurious DMA transfers.

In write-posting mode, the DMA or the MPU fill the FIFO with no consideration to the associated byte enables. Any byte stored in the FIFO is written into the memory device.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.12.4.6 Optimizing NAND Access Using the Prefetch and Write-Posting Engine

Access time to a NAND memory device can be optimized for back-to-back accesses if the associated $\overline{CS}$ signal is not deasserted between accesses. The GPMC access engine can track prefetch engine accesses to optimize the access timing parameter programmed for the allocated chip-select, if no accesses to other chip-selects (that is, interleaved accesses) occur. Similarly, the access engine also eliminates the CYCLE2CYCLEDELAY even if CYCLE2CYCLESAMECSEN is set. This capability is limited to the prefetch and write-posting engine accesses, and MPU accesses to a NAND memory device (through the defined chip-select memory region or through the GPMC_NAND_DATA_i(where , i = 0 to 7) are never optimized.

The GPMC_PREFETCH_CONFIG1[27] ENABLEOPTIMIZEDACCESS bit must be set to enable optimized accesses. To optimize access time, the GPMC_PREFETCH_CONFIG1[30-28] CYCLEOPTIMIZATION field defines the number of GPMC_FCLK cycles to be suppressed from the RDCYCLETIME, WRCYCLETIME, RDACCESSTIME, WRACCESSTIME, CSOFFTIME, ADVOFFTIME, OEOFFTIME, and WEOFFTIME timing parameters.

Figure 6-42, in the case of back-to-back accesses to the NAND flash through the prefetch engine, CYCLE2CYCLESAMECSEN is forced to 0 when using optimized accesses. The first access uses the regular timing settings for this chip-select. All accesses after this one use settings reduced by x clock cycles, x being defined by the GPMC_PREFETCH_CONFIG1[30-28] CYCLEOPTIMIZATION field.

Figure 6-42. NAND Read Cycle Optimization Timing Description

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.4.12.4.7 Interleaved Accesses Between Prefetch and Write-Posting Engine and Other Chip-Selects

Any on-going read or write access from the prefetch and write-posting engine is completed before an access to any other chip-select can be initiated. As a default, the arbiter uses a fixed-priority algorithm, and the prefetch and write-posting engine has the lowest priority. The maximum latency added to access starting time in this case equals the RDCYCLETIME or WRCYCLETIME (optimized or not) plus the requested BUSTURNAROUND delay for bus turnaround completion programmed for the chip-select to which the NAND device is connected to.

Alternatively, a round-robin arbitration can be used to prioritize accesses to the external bus. This arbitration scheme is enabled by setting the GPMC_PREFETCH_CONFIG1[23] PFPWENROUNDROBIN bit. When a request to another chip-select is received while the prefetch and write-posting engine is active, priority is given to the new request. The request processed thereafter is the prefetch and write-posting engine request, even if another interconnect request is passed in the mean time. The engine keeps control of the bus for an additional number of requests programmed in the GPMC_PREFETCH_CONFIG1[19-16] PFPWWEIGHTEDPRIO bit field. Control is then passed to the direct interconnect request.

As an example, the round-robin arbitration scheme is selected with PFPWWEIGHTEDPRIO set to 2h. Considering the prefetch and write-posting engine and the interconnect interface are always requesting access to the external interface, the GPMC grants priority to the direct interconnect access for one request. The GPMC then grants priority to the engine for three requests, and finnaly back to the direct interconnect access, until the arbiter is reset when one of the two initiators stops initiating requests.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.5 GPMC High-Level Programming Model Overview

The high-level programming model introduces a top-down approach for users that need to configure the GPMC module. Figure 6-43 shows a programming model top-level diagram for the GPMC. Each block of the diagram is described in one of the following subsections through a set of registers to configure. Table 6-28 and Table 6-29 list each step in the model.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 6-43. Programming Model Top-Level Diagram

Copyright © 2011, Texas Instruments Incorporated

Table 6-28. GPMC Configuration in NOR Mode

Step	Description
NOR Memory Type	See Table 6-31
NOR Chip-Select Configuration	See Table 6-32
NOR Timings Configuration	See Table 6-33
Wait Pin Configuration	See Table 6-34
Enable Chip-Select	See Table 6-35

Table 6-29. GPMC Configuration in NAND Mode

Step	Description
NAND Memory Type	See Table 6-36
NAND Chip-Select Configuration	See Table 6-37
Write Operations (Asynchronous)	See Table 6-38
Read Operations (Asynchronous)	See Table 6-38
ECC Engine	See Table 6-39
Prefetch and Write-Posting Engine	See Table 6-40
Wait Pin Configuration	See Table 6-41
Enable Chip-Select	See Table 6-42

6.1.3.6 GPMC Initialization

Table 6-30 describes the settings required to reset the GPMC.

Table 6-30. Reset GPMC

Sub-process Name	Register / Bitfield	Value
Start a software reset	GPMC_SYSCONFIG[1] SOFTRESET	1
Wait until	GPMC_SYSSTATUS[0] RESETDONE	1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.7 GPMC Configuration in NOR Mode

This section gives a generic configuration for parameters related to the NOR memory connected to the GPMC. Table 6-31 through Table 6-35 list the steps to configure the GPMC in NOR mode.

NOTE: In the tables of this section, 'x' in Value column stands for 'depends on configuration'.

Table 6-31. NOR Memory Type

Sub-process Name	Register / Bitfield	Value
Set the NOR protocol	GPMC_CONFIG1_i[11-10] DEVICETYPE	0
Set a device size	GPMC_CONFIG1_i[13-12] DEVICESIZE	x
Select an address and data multiplexing protocol	GPMC_CONFIG1_i[9] MUXADDDATA	x
Set the attached device page length	GPMC_CONFIG1_i[24-23] ATTACHEDDEVICEPAGELENGTH	x
Set the wrapping burst capabilities	GPMC_CONFIG1_i[31] WRAPBURST	x
Select a timing signals latencies factor	GPMC_CONFIG1_i[4] TIMEPARAGRANULARITY	x
Select an output clock frequency	GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER	x
Choose an output clock activation time	GPMC_CONFIG1_i[26-25] CLKACTIVATIONTIME	x
Set a single or multiple access for read operations	GPMC_CONFIG1_i[30] READMULTIPLE	x
Set a synchronous or asynchronous mode for read operations	GPMC_CONFIG1_i[29] READTYPE	x
Set a single or multiple access for write operations	GPMC_CONFIG1_i[28] WRITEMULTIPLE	x
Set a synchronous or asynchronous mode for write operations	GPMC_CONFIG1_i[27] WRITETYPE	x

Table 6-32. NOR Chip-Select Configuration

Sub-process Name	Register / Bitfield	Value
Select the chip-select base address	GPMC_CONFIG7_i[5-0] BASEADDRESS	x
Select the chip-select mask address	GPMC_CONFIG7_i[11-8] MASKADDRESS	x

Table 6-33. NOR Timings Configuration

Sub-process Name	Register / Bitfield	Value
Configure adequate timing parameters in various memory modes	See Section 6.1.3.10	

Table 6-34. WAIT Pin Configuration

Sub-process Name	Register / Bitfield	Value
Enable or disable wait pin monitoring for read operations	GPMC_CONFIG1_i[22] WAITREADMONITORING	x
Enable or disable wait pin monitoring for write operations	GPMC_CONFIG1_i[21] WAITWRITEMONITORING	x
Select a wait pin monitoring time	GPMC_CONFIG1_i[19-18] WAITMONITORINGTIME	x
Choose the input wait pin for the chip-select	GPMC_CONFIG1_i[17-16] WAITPINSELECT	x

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 6-35. Enable Chip-Select

Sub-process Name	Register / Bitfield	Value
When all parameters are configured, enable the chip-select	GPMC_CONFIG7_i[6] CSVALID	x

6.1.3.8 GPMC Configuration in NAND Mode

This section gives a generic configuration for parameters related to NAND memory connected to the GPMC.

Table 6-36. NAND Memory Type

Sub-process Name	Register / Bitfield	Value
Set the NAND protocol	GPMC_CONFIG1_i[11-10] DEVICETYPE	2h
Set a device size	GPMC_CONFIG1_i[13-12] DEVICESIZE	x
Set the address and data multiplexing protocol to non-multiplexed attached device	GPMC_CONFIG1_i[9] MUXADDDATA	0
Select a timing signals latencies factor	GPMC_CONFIG1_i[4] TIMEPARAGRANULARITY	x
Set a synchronous or asynchronous mode and a single or multiple access for read and write operations	See Section 6.1.3.9	x

Table 6-37. NAND Chip-Select Configuration

Sub-process Name	Register / Bitfield	Value
Select the chip-select base address	GPMC_CONFIG7_i[5-0] BASEADDRESS	x
Select the chip-select minimum granularity (16M bytes)	GPMC_CONFIG7_i[11-8] MASKADDRESS	x

Table 6-38. Asynchronous Read and Write Operations

Sub-process Name	Register / Bitfield	Value
Configure adequate timing parameters in asynchronous modes	See Section 6.1.3.10	

Table 6-39. ECC Engine

Sub-process Name	Register / Bitfield	Value
Select the ECC result register where the first ECC computation is stored (Only applies to Hamming)	GPMC_ECC_CONTROL[3-0] ECCPOINTER	x
Clear all ECC result registers	GPMC_ECC_CONTROL[8] ECCCLEAR	Write 1 to clear
Define ECCSIZE0 and ECCSIZE1	GPMC_ECC_SIZE_CONFIG[19-12] ECCSIZE0 and GPMC_ECC_SIZE_CONFIG[29-22] ECCSIZE1	x
Select the size of each of the 9 result registers (size specified by ECCSIZE0 or ECCSIZE1)	GPMC_ECC_SIZE_CONFIG[j-1] ECCjRESULTSIZE where j = 1 to 9	x
Select the chip-select where ECC is computed	GPMC_ECC_SIZE_CONFIG[3-1] ECCCS	x
Select the Hamming code or BCH code ECC algorithm in use	GPMC_ECC_SIZE_CONFIG[16] ECCALGORITHM	x
Select word size for ECC calculation	GPMC_ECC_SIZE_CONFIG[7] ECC16B	x
If the BCH code is used, Set an error correction capability and Select a number of sectors to process	GPMC_ECC_SIZE_CONFIG[13-12] ECCBCHTSEL and GPMC_ECC_SIZE_CONFIG[6-4] ECCTOPSECTOR	x

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 6-39. ECC Engine (continued)

Sub-process Name	Register / Bitfield	Value
Enable the ECC computation	GPMC_ECC_SIZE_CONFIG[0] ECCENABLE	1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 6-40. Prefetch and Write-Posting Engine

Sub-process Name	Register / Bitfield	Value
Disable the engine before configuration	GPMC_PREFETCH_CONTROL[0] STARTENGINE	0
Select the chip-select associated with a NAND device where the prefetch engine is active	GPMC_PREFETCH_CONFIG1[26-24] ENGINECSSELECTOR	x
Select access direction through prefetch engine, read or write.	GPMC_PREFETCH_CONFIG1[0] ACCESSMODE	x
Select the threshold used to issue a DMA request	GPMC_PREFETCH_CONFIG1[14-8] FIFOTHRESHOLD	x
Select either DMA synchronized mode or SW manual mode.	GPMC_PREFETCH_CONFIG1[2] DMAMODE	x
Select if the engine immediately starts accessing the memory upon STARTENGINE assertion or if hardware synchronization based on a WAIT signal is used.	GPMC_PREFETCH_CONFIG1[3] SYNCHROMODE	x
Select which wait pin edge detector should start the engine in synchronized mode	GPMC_PREFETCH_CONFIG1[5-4] WAITPINSELECTOR	x
Enter a number of clock cycles removed to timing parameters (For all back-to-back accesses to the NAND flash but not the first one)	GPMC_PREFETCH_CONFIG1[30-28] CYCLEOPTIMIZATION	x
Enable the prefetch postwite engine	GPMC_PREFETCH_CONFIG1[7] ENABLEENGINE	1
Select the number of bytes to be read or written by the engine to the selected chip-select	GPMC_PREFETCH_CONFIG2[13-0] TRANSFERCOUNT	x
Start the prefetch engine	GPMC_PREFETCH_CONTROL[0] STARTENGINE	1

Table 6-41. WAIT Pin Configuration

Sub-process Name	Register / Bitfield	Value
Selects when the engine starts the access to CS	GPMC_PREFETCH_CONFIG1[3] SYNCHROMODE	x
Select which wait pin edge detector should start the engine in synchronized mode	GPMC_PREFETCH_CONFIG1[5-4] WAITPINSELECTOR	x

Table 6-42. Enable Chip-Select

Sub-process Name	Register / Bitfield	Value
When all parameters are configured, enable the chip-select	GPMC_CONFIG7_i[6] CSVALID	x

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.9 Set Memory Access

This section details the bit field to configure to set the GPMC in various memory modes.

Table 6-43. Mode Parameters Check List Table

Register	Bit	Bit Field Name	Asynchronous				Synchronous			
			Single Read Access	Single Write Access	Multiple Read (Page) Access	Multiple Write (Page) Access	Single Read Access	Single Write Access	Multiple Read (Burst) Access	Multiple Write (Burst) Access
GPMC_CONFIG1_i	30	READMULTIPLE	0	-	1	N/S	0	-	1	-
GPMC_CONFIG1_i	29	READTYPE	0	-	0	N/S	1	-	1	-
GPMC_CONFIG1_i	28	WRITEMULTIPLE	-	0		N/S	-	0	-	1
GPMC_CONFIG1_i	27	WRITETYPE	-	0		N/S	-	1	-	1

Table 6-44. Access Type Parameters Check List Table

Register	Bit	Bit Field Name	Access Type		
			Non-Mux	Address/Data Mux	AAD Mux
GPMC_CONFIG1_i	9-8	MUXADDDATA	0	2h	1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.10 GPMC Timing Parameters

Figure 6-44 shows a programming model diagram for the NOR interfacing timing parameters.

Table 6-45 lists bit fields to configure adequate timing parameter in various memory modes.

Figure 6-44. NOR Interfacing Timing Parameters Diagram

DRAFT

Copyright © 2011, Texas Instruments Incorporated

Table 6-45. Timing Parameters

Register	Bit	Bit Field Name	Asynchronous			Synchronous				Access Type		
			Single Read Access	Single Write Access	Multiple Read (Page) access	Single Read Access	Single Write Access	Multiple Read (Burst) Access	Multiple Write (Burst) Access	Non-multiplexed	Address /Data-multiplexed	AAD-multiplexed
GPMC_CONFIG1_i	9-8	MUXADDDATA	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG1_i	29	READTYPE	y		y	y		y		y	y	y
GPMC_CONFIG1_i	30	READMULTIPLE	y		y	y		y		y	y	y
GPMC_CONFIG1_i	27	WRITETYPE		y			y		y	y	y	y
GPMC_CONFIG1_i	28	WRITEMULTIPLE		y			y		y	y	y	y
GPMC_CONFIG1_i	31	WRAPBURST						y	y	y	y	y
GPMC_CONFIG1_i	26-25	CLKACTIVATIONTIME				y	y	y	y	y	y	y
GPMC_CONFIG1_i	19-18	WAITMONITORINGTIME	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG1_i	4	TIMEPARAGRANULARITY	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG2_i	20-16	CSWROFFTIME		y			y		y	y	y	y
GPMC_CONFIG2_i	12-8	CSRDOFFTIME	y		y	y		y		y	y	y
GPMC_CONFIG2_i	7	CSEXTRADELAY	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG2_i	3-0	CSONTIME	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG3_i	30-28	ADVAADMUXWROFFTIME		y			y		y			y
GPMC_CONFIG3_i	26-24	ADVAADMUXRDOFFTIME	y		y	y		y				y
GPMC_CONFIG3_i	6-4	ADVAADMUXONTIME	y	y	y	y	y	y	y			y
GPMC_CONFIG3_i	20-16	ADVWROFFTIME		y			y		y	y	y	y
GPMC_CONFIG3_i	12-8	ADVRDOFFTIME	y		y	y		y		y	y	y
GPMC_CONFIG3_i	7	ADVEXTRADELAY	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG3_i	3-0	ADVONTIME	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG4_i	15-13	OEAADMUXOFFTIME	y	y	y	y	y	y	y			y
GPMC_CONFIG4_i	6-4	OEAADMUXONTIME	y	y	y	y	y	y	y			y
GPMC_CONFIG4_i	28-24	WEOFFTIME		y			y		y	y	y	y
GPMC_CONFIG4_i	23	WEEXTRADELAY		y			y		y	y	y	y
GPMC_CONFIG4_i	19-16	WEONTIME		y			y		y	y	y	y
GPMC_CONFIG4_i	12-8	OEOFFTIME	y		y	y		y		y	y	y
GPMC_CONFIG4_i	7	OEEXTRADELAY	y		y	y		y		y	y	y
GPMC_CONFIG4_i	3-0	OEONTIME	y		y	y		y		y	y	y
GPMC_CONFIG5_i	27-24	PAGEBURSTACCESSTIME			y			y	y	y	y	y
GPMC_CONFIG5_i	20-16	RDACCESSTIME	y		y	y		y		y	y	y
GPMC_CONFIG5_i	12-8	WRCYCLETIME		y			y		y	y	y	y
GPMC_CONFIG5_i	4-0	RDCYCLETIME	y		y	y		y		y	y	y
GPMC_CONFIG6_i	28-24	WRACCESSTIME		y			y		y	y	y	y
GPMC_CONFIG6_i	19-16	WRDATAONADMUXBUS		y			y		y		y	y

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 6-45. Timing Parameters (continued)

Register	Bit	Bit Field Name	Asynchronous			Synchronous				Access Type		
			Single Read Access	Single Write Access	Multiple Read (Page) access	Single Read Access	Single Write Access	Multiple Read (Burst) Access	Multiple Write (Burst) Access	Non-multiplexed	Address /Data-multiplexed	AAD-multiplexed
GPMC_CONFIG6_i	11-8	CYCLE2CYCLEDELAY	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG6_i	7	CYCLE2CYCLESAMECSEN	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG6_i	6	CYCLE2CYCLEDIFFCSEN	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG6_i	3-0	BUSTURNAROUND	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG7_i	6	CSVALID	y	y	y	y	y	y	y	y	y	y

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.3.10.1 GPMC Timing Parameters Formulas

This section intends to help the user to calculate the GPMC timing bit fields values. Formulas are not listed exhaustively.

The section details:

- NAND Flash Interface Timing Parameters Formulas
- Synchronous NOR Flash Timing Parameters Formulas
- Asynchronous NOR Flash Timing Parameters Formulas

6.1.3.10.1.1 NAND Flash Interface Timing Parameters Formulas

This section lists formulas to use in order to calculate NAND timing parameters. This is the case when GPMC_CONFIG1_i[11-10] DEVICETYPE = 2h. Table 6-46 details NAND timing parameters.

Table 6-46. NAND Formulas Description Table

Configuration Parameter	Unit	Description
A	ns	Pulse duration - $\overline{\text{GPMC_WE}}$ valid time
B	ns	Delay time - $\overline{\text{GPMC_CS}}$ valid to $\overline{\text{GPMC_WE}}$ valid
C	ns	Delay time - GPMC_$\overline{\text{BE0}}$_CLE/GPMC_$\overline{\text{ADV}}$_ALE high to $\overline{\text{GPMC_WE}}$ valid
D	ns	Delay time - GPMC_AD[15:0] valid to $\overline{\text{GPMC_WE}}$ valid
E	ns	Delay time - $\overline{\text{GPMC_WE}}$ invalid to GPMC_AD[15:0] invalid
F	ns	Delay time - $\overline{\text{GPMC_WE}}$ invalid to GPMC_$\overline{\text{BE0}}$_CLE/GPMC_$\overline{\text{ADV}}$_ALE invalid
G	ns	Delay time - $\overline{\text{GPMC_WE}}$ invalid to $\overline{\text{GPMC_CS}}$ invalid
H	ns	Cycle time - Write cycle time
I	ns	Delay time - $\overline{\text{GPMC_CS}}$ valid to $\overline{\text{GPMC_OE}}$ valid
J	ns	Setup time - GPMC_AD[15:0] valid to $\overline{\text{GPMC_OE}}$ invalid
K	ns	Pulse duration - $\overline{\text{GPMC_OE}}$ valid time
L	ns	Cycle time - Read cycle time
M	ns	Delay time - $\overline{\text{GPMC_OE}}$ invalid to $\overline{\text{GPMC_CS}}$ invalid

The configuration parameters are calculated through the following formulas.

A = (WEOffTime - WEOnTime) * (TimeParaGranularity + 1) * GPMC_FCLK period

B = ((WEOnTime - CSOnTime) * (TimeParaGranularity + 1) + 0.5 * (WEExtraDelay - CSExtraDelay)) * GPMC_FCLK period

C = ((WEOnTime - ADVOnTime) * (TimeParaGranularity + 1) + 0.5 * (WEExtraDelay - ADVExtraDelay)) * GPMC_FCLK period

D = (WEOnTime * (TimeParaGranularity + 1) + 0.5 * WEExtraDelay) * GPMC_FCLK period

E = (WrCycleTime - WEOffTime * (TimeParaGranularity + 1) - 0.5 * WEExtraDelay) * GPMC_FCLK period

F = (ADVWrOffTime - WEOffTime * (TimeParaGranularity + 1) + 0.5 * (ADVExtraDelay - WEExtraDelay) * GPMC_FCLK period

G = (CSWrOffTime - WEOffTime * (TimeParaGranularity + 1) + 0.5 * (CSExtraDelay - WEExtraDelay) * GPMC_FCLK period

H = WrCycleTime * (1 + TimeParaGranularity) * GPMC_FCLK period

I = ((OEOnTime - CSOnTime) * (TimeParaGranularity + 1) + 0.5 * (OEExtraDelay - CSExtraDelay)) * GPMC_FCLK period

J = ((AccessTime - OEOffTime) * (TimeParaGranularity + 1) - 0.5 * OEExtraDelay)) * GPMC_FCLK period

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

K = (OEOffTime - OEOnTime) * (1 + TimeParaGranularity) * GPMC_FCLK period

L = RdCycleTime * (1 + TimeParaGranularity) * GPMC_FCLK period

M = (CSRdOffTime - OEOffTime * (TimeParaGranularity + 1) + 0.5 * (CSExtraDelay - OEExtraDelay) * GPMC_FCLK period

Figure 6-45 shows a command latch cycle timing simplified example where formulas are associated with signal waves.

Figure 6-45. NAND Command Latch Cycle Timing Simplified Example

6.1.3.10.1.2 Synchronous NOR Flash Timing Parameters Formulas

This section lists all formulas to use in order to calculate synchronous NOR timing parameters. This is the case when GPMC_CONFIG1_i[11-10] DEVICETYPE = 0 and when READTYPE or WRITETYPE are set to synchronous mode.

Table 6-47. Synchronous NOR Formulas Description Table

Configuration Parameter	Unit	Description
A	ns	Pulse duration - $\overline{\text{GPMC_CS}}$ low
B	ns	Delay time - address bus valid to GPMC_CLK first edge
		Delay time - GPMC_$\overline{\text{BE0}}$_CLE/$\overline{\text{GPMC_BE1}}$ valid to GPMC_CLK first edge
C	ns	Pulse duration - GPMC_$\overline{\text{BE0}}$_CLE/$\overline{\text{GPMC_BE1}}$ low
D	ns	Delay time - GPMC_CLK rising edge to GPMC_$\overline{\text{BE0}}$_CLE/$\overline{\text{GPMC_BE1}}$ invalid
		Delay time - GPMC_CLK rising edge to GPMC_$\overline{\text{ADV}}$_ALE invalid
E	ns	Delay time - GPMC_CLK rising edge to $\overline{\text{GPMC_CS}}$ invalid
		Delay time - GPMC_CLK rising edge to $\overline{\text{GPMC_OE}}$ invalid
F	ns	Delay time - GPMC_CLK rising edge to $\overline{\text{GPMC_CS}}$ transition
G	ns	Delay time - GPMC_CLK rising edge to GPMC_$\overline{\text{ADV}}$_ALE transition
H	ns	Delay time - GPMC_CLK rising edge to $\overline{\text{GPMC_OE}}$ transition
I	ns	Delay time - GPMC_CLK rising edge to $\overline{\text{GPMC_WE}}$ transition
J	ns	Delay time - GPMC_CLK rising edge to GPMC_AD data bus transition
		Delay time - GPMC_CLK rising edge to GPMC_$\overline{\text{BE0}}$_CLE/$\overline{\text{GPMC_BE1}}$ transition
K	ns	Pulse duration - GPMC_$\overline{\text{ADV}}$_ALE low
L	ns	Delay time - GPMC_WAIT invalid to first data latching GPMC_CLK edge

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

The configuration parameters are calculated through the following formulas.

For single read accesses:

A = (CSRDOFFTIME - CSONTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period

C = RDCYCLETIME * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period

D = (RDCYCLETIME - ACCESSTIME) * GPMC_FCLK period

E = (CSRDOFFTIME - ACCESSTIME) * GPMC_FCLK period

For burst read accesses (where n is the page burst access number):

A = (CSRDOFFTIME - CSONTIME + (n - 1) * PAGEBURSTACCESSTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period

C = (RDCYCLETIME + (n - 1) * PAGEBURSTACCESSTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period

D = (RDCYCLETIME - (ACCESSTIME + (n - 1) * PAGEBURSTACCESSTIME) * GPMC_FCLK period

E = (CSRDOFFTIME - (ACCESSTIME + (n - 1) * PAGEBURSTACCESSTIME) * GPMC_FCLK period

For burst write accesses (where n is the page burst access number):

A = (CSWROFFTIME - CSONTIME + (n - 1) * PAGEBURSTACCESSTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period

C = (WRCYCLETIME + (n - 1) * PAGEBURSTACCESSTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period

D = (WRCYCLETIME - (ACCESSTIME + (n - 1) * PAGEBURSTACCESSTIME) * GPMC_FCLK period

E = (CSWROFFTIME - (ACCESSTIME + (n - 1) * PAGEBURSTACCESSTIME) * GPMC_FCLK period

For all accesses:

For $\overline{CS}$ falling edge ($\overline{CS}$ activated):

- Case where GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER = 0x0

  F = 0.5 * CSEXTRADELAY * GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 0x1

  F = 0.5 * CSEXTRADELAY * GPMC_FCLK period, when (CLKACTIVATIONTIME and CSONTIME are odd) or (CLKACTIVATIONTIME and CSONTIME are even)

  F = (1 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period otherwise
- Case where GPMCFCLKDIVIDER = 0x2

  F = 0.5 * CSEXTRADELAY * GPMC_FCLK period, when (CSONTIME - CLKACTIVATIONTIME) is a multiple of 3

  F = (1 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSONTIME - CLKACTIVATIONTIME - 1) is a multiple of 3

  F = (2 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSONTIME - CLKACTIVATIONTIME - 2) is a multiple of 3

For $\overline{CS}$ rising edge ($\overline{CS}$ de-activated) in reading mode:

- Case where GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER = 0x0

  F = 0.5 * CSEXTRADELAY * GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 0x1

  F = 0.5 * CSEXTRADELAY * GPMC_FCLK period, when (CLKACTIVATIONTIME and CSRDOFFTIME are odd) or (CLKACTIVATIONTIME and CSRDOFFTIME are even)

  F = (1 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period otherwise
- Case where GPMCFCLKDIVIDER = 0x2

  F = 0.5 * CSEXTRADELAY * GPMC_FCLK period, when (CSRDOFFTIME - CLKACTIVATIONTIME) is a multiple of 3

  F = (1 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSRDOFFTIME - CLKACTIVATIONTIME - 1) is a multiple of 3

  F = (2 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSRDOFFTIME -

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

CLKACTIVATIONTIME - 2) is a multiple of 3

For $\overline{CS}$ rising edge ($\overline{CS}$ de-activated) in writing mode:

- Case where GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER = 0x0

  F = 0.5 * CSEXTRADELAY * GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 0x1

  F = 0.5 * CSEXTRADELAY * GPMC_FCLK period, when (CLKACTIVATIONTIME and CSWROFFTIME are odd) or (CLKACTIVATIONTIME and CSWROFFTIME are even)

  F = (1 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period otherwise
- Case where GPMCFCLKDIVIDER = 0x2

  F = 0.5 * CSEXTRADELAY * GPMC_FCLK period, when (CSWROFFTIME - CLKACTIVATIONTIME) is a multiple of 3

  F = (1 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSWROFFTIME - CLKACTIVATIONTIME - 1) is a multiple of 3

  F = (2 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSWROFFTIME - CLKACTIVATIONTIME - 2) is a multiple of 3

For $\overline{ADV}$ falling edge ($\overline{ADV}$ activated):

- Case where GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER = 0x0

  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 0x1

  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period, when (CLKACTIVATIONTIME and ADVONTIME are odd) or (CLKACTIVATIONTIME and ADVONTIME are even)

  G = (1 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period otherwise
- Case where GPMCFCLKDIVIDER = 0x2

  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period, when (ADVONTIME - CLKACTIVATIONTIME) is a multiple of 3

  G = (1 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVONTIME - CLKACTIVATIONTIME - 1) is a multiple of 3

  G = (2 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVONTIME - CLKACTIVATIONTIME - 2) is a multiple of 3

For $\overline{ADV}$ rising edge ($\overline{ADV}$ de-activated) in reading mode:

- Case where [1-0] GPMCFCLKDIVIDER = 0x0

  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 0x1

  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period, when (CLKACTIVATIONTIME and ADVRDOFFTIME are odd) or (CLKACTIVATIONTIME and ADVRDOFFTIME are even)

  G = (1 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period otherwise
- Case where GPMCFCLKDIVIDER = 0x2

  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period, when (ADVRDOFFTIME - CLKACTIVATIONTIME) is a multiple of 3

  G = (1 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVRDOFFTIME - CLKACTIVATIONTIME - 1) is a multiple of 3

  G = (2 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVRDOFFTIME - CLKACTIVATIONTIME - 2) is a multiple of 3

For $\overline{ADV}$ rising edge ($\overline{ADV}$ de-activated) in writing mode:

- Case where [1-0] GPMCFCLKDIVIDER = 0x0

  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 0x1

  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period, when (CLKACTIVATIONTIME and

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

ADVWROFFTIME are odd) or (CLKACTIVATIONTIME and ADVWROFFTIME are even)

G = (1 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period otherwise

- Case where GPMCFCLKDIVIDER = 0x2

  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period, when (ADVWROFFTIME - CLKACTIVATIONTIME) is a multiple of 3

  G = (1 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVWROFFTIME - CLKACTIVATIONTIME - 1) is a multiple of 3

  G = (2 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVWROFFTIME - CLKACTIVATIONTIME - 2) is a multiple of 3

For $\overline{OE}$ falling edge ($\overline{OE}$ activated):

- Case where GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER = 0x0

  H = 0.5 * OEEXTRADELAY * GPMC_FCLK period

- Case where GPMCFCLKDIVIDER = 0x1

  H = 0.5 * OEEXTRADELAY * GPMC_FCLK period, when (CLKACTIVATIONTIME and OEONTIME are odd) or (CLKACTIVATIONTIME and OEONTIME are even)

  H = (1 + 0.5 * OEEXTRADELAY) * GPMC_FCLK period otherwise

- Case where GPMCFCLKDIVIDER = 0x2

  H = 0.5 * OEEXTRADELAY * GPMC_FCLK period, when (OEONTIME - CLKACTIVATIONTIME) is a multiple of 3

  H = (1 + 0.5 * OEEXTRADELAY) * GPMC_FCLK period, when (OEONTIME - CLKACTIVATIONTIME - 1) is a multiple of 3

  H = (2 + 0.5 * OEEXTRADELAY) * GPMC_FCLK period, when (OEONTIME - CLKACTIVATIONTIME - 2) is a multiple of 3

For $\overline{OE}$ rising edge ($\overline{OE}$ de-activated):

- Case where [1-0] GPMCFCLKDIVIDER = 0x0

  H = 0.5 * OEEXTRADELAY * GPMC_FCLK period

- Case where GPMCFCLKDIVIDER = 0x1

  H = 0.5 * OEEXTRADELAY * GPMC_FCLK period, when (CLKACTIVATIONTIME and OEOFFTIME are odd) or (CLKACTIVATIONTIME and OEOFFTIME are even)

  H = (1 + 0.5 * OEEXTRADELAY) * GPMC_FCLK period otherwise

- Case where GPMCFCLKDIVIDER = 0x2

  H = 0.5 * OEEXTRADELAY * GPMC_FCLK period, when (OEOFFTIME - CLKACTIVATIONTIME) is a multiple of 3

  H = (1 + 0.5 * OEEXTRADELAY) * GPMC_FCLK period, when (OEOFFTIME - CLKACTIVATIONTIME - 1) is a multiple of 3

  H = (2 + 0.5 * OEEXTRADELAY) * GPMC_FCLK period, when (OEOFFTIME - CLKACTIVATIONTIME - 2) is a multiple of 3

For $\overline{WE}$ falling edge ($\overline{WE}$ activated):

- Case where GPMC_CONFIG1_i[1-0] GPMCFCLKDIVIDER = 0x0

  I = 0.5 * WEEXTRADELAY * GPMC_FCLK period

- Case where GPMCFCLKDIVIDER = 0x1

  I = 0.5 * WEEXTRADELAY * GPMC_FCLK period, when (CLKACTIVATIONTIME and WEONTIME are odd) or (CLKACTIVATIONTIME and WEONTIME are even)

  I = (1 + 0.5 * WEEXTRADELAY) * GPMC_FCLK period otherwise

- Case where GPMCFCLKDIVIDER = 0x2

  I = 0.5 * WEEXTRADELAY * GPMC_FCLK period, when (WEONTIME - CLKACTIVATIONTIME) is a multiple of 3

  I = (1 + 0.5 * WEEXTRADELAY) * GPMC_FCLK period, when (WEONTIME - CLKACTIVATIONTIME -

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

1) is a multiple of 3

I = (2 + 0.5 * WEEXTRADELAY) * GPMC_FCLK period, when (WEONTIME - CLKACTIVATIONTIME - 2) is a multiple of 3

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

For $\overline{WE}$ rising edge ($\overline{WE}$ de-activated):

- Case where [1-0] GPMCFCLKDIVIDER = 0x0

  I = 0.5 * WEEXTRADELAY * GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 0x1

  I = 0.5 * WEEXTRADELAY * GPMC_FCLK period, when (CLKACTIVATIONTIME and WEOFFTIME are odd) or (CLKACTIVATIONTIME and WEOFFTIME are even)

  I = (1 + 0.5 * WEEXTRADELAY) * GPMC_FCLK period otherwise
- Case where GPMCFCLKDIVIDER = 0x2

  I = 0.5 * WEEXTRADELAY * GPMC_FCLK period, when (WEOFFTIME - CLKACTIVATIONTIME) is a multiple of 3

  I = (1 + 0.5 * WEEXTRADELAY) * GPMC_FCLK period, when (WEOFFTIME - CLKACTIVATIONTIME - 1) is a multiple of 3

  I = (2 + 0.5 * WEEXTRADELAY) * GPMC_FCLK period, when (WEOFFTIME - CLKACTIVATIONTIME - 2) is a multiple of 3

For GPMC_$\overline{ADV}$ low pulse duration:

- Read operation

  K = (ADVRDOFFTIME - ADVONTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period
- Write operation

  K = (ADVWROFFTIME - ADVONTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period

For GPMC_WAIT invalid to first data latching GPMC_CLK edge:

L = WAITMONITORINGTIME * (GPMCFCLKDIVIDER + 1) * GPMC_FCLK period + GPMC_CLK period

Figure 6-46 shows a synchronous NOR single read simplified example where formulas are associated with signal waves.

Figure 6-46. Synchronous NOR Single Read Simplified Example

Copyright © 2011, Texas Instruments Incorporated

6.1.3.10.1.3 Asynchronous NOR Flash Timing Parameters Formulas

This section lists all formulas to use in order to calculate asynchronous NOR timing parameters. This is the case when [11-10] DEVICETYPE = 0x0 and when READTYPE or WRITETYPE are set to asynchronous mode.

Table 6-48. Asynchronous NOR Formulas Description Table

Configuration Parameter	Unit	Description
A	ns	Pulse duration - $\overline{\text{GPMC_CS}}$ low
B	ns	Delay time - $\overline{\text{GPMC_CS}}$ valid to GPMC_$\overline{\text{ADV}}$_ALE invalid
C	ns	Delay time - $\overline{\text{GPMC_CS}}$ valid to $\overline{\text{GPMC_OE}}$ invalid (single read)
D	ns	Pulse duration - address bus valid - 2nd, 3rd and 4th accesses
E	ns	Delay time - $\overline{\text{GPMC_CS}}$ valid to $\overline{\text{GPMC_WE}}$ valid
F	ns	Delay time - $\overline{\text{GPMC_CS}}$ valid to $\overline{\text{GPMC_WE}}$ invalid
G	ns	Address invalid duration between 2 successive R/W accesses
H	ns	Setup time - read data valid before $\overline{\text{GPMC_OE}}$ high
I	ns	Delay time - $\overline{\text{GPMC_CS}}$ valid to $\overline{\text{GPMC_OE}}$ invalid (burst read)
J	ns	Delay time - address bus valid to $\overline{\text{GPMC_CS}}$ valid
		Delay time - data bus valid to $\overline{\text{GPMC_CS}}$ valid
		Delay time - GPMC_$\overline{\text{BE0}}$_CLE/$\overline{\text{GPMC_BE1}}$ valid to $\overline{\text{GPMC_CS}}$ valid
K	ns	Delay time - $\overline{\text{GPMC_CS}}$ valid to GPMC_$\overline{\text{ADV}}$_ALE valid
L	ns	Delay time - $\overline{\text{GPMC_CS}}$ valid to $\overline{\text{GPMC_OE}}$ valid
M	ns	Delay time - GPMC_CS valid to first data latching edge
N	ns	Pulse duration - GPMC_$\overline{\text{BE0}}$_CLE/$\overline{\text{GPMC_BE1}}$ valid time
O	ns	Delay time - $\overline{\text{GPMC_CS}}$ valid to GPMC_$\overline{\text{ADV}}$_ALE valid

The configuration parameters are calculated through the following formulas. Note that these formulas are not exhaustive.

$\overline{\text{GPMC_CS}}$ low pulse:

For single read: A = (CSRDOFFTIME - CSONTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period

For burst read: A = (CSRDOFFTIME - CSONTIME + (N - 1) * PAGEBURSTACCESSTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period, where N = page burst access number

For single write: A = (CSWROFFTIME - CSONTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period

For burst write: A = (CSWROFFTIME - CSONTIME + (N - 1) * PAGEBURSTACCESSTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period, where N = page burst access number

$\overline{\text{GPMC_CS}}$ valid to GPMC_$\overline{\text{ADV}}$_ALE invalid delay:

For reading: B = ((ADVRDOFFTIME - CSONTIME) * (TIMEPARAGRANULARITY + 1) + 0.5 * (ADVEXTRADELAY - CSEXTRADELAY)) * GPMC_FCLK period

For writing: B = ((ADVWROFFTIME - CSONTIME) * (TIMEPARAGRANULARITY + 1) + 0.5 * (ADVEXTRADELAY - CSEXTRADELAY)) * GPMC_FCLK period

C = ((OEOFFTIME - CSONTIME) * (TIMEPARAGRANULARITY + 1) + 0.5 * (OEEXTRADELAY - CSEXTRADELAY)) * GPMC_FCLK period

D = PAGEBURSTACCESSTIME * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period

E = ((WEONTIME - CSONTIME) * (TIMEPARAGRANULARITY + 1) + 0.5 * (WEEXTRADELAY - CSEXTRADELAY)) * GPMC_FCLK period

F = ((WEOFFTIME - CSONTIME) * (TIMEPARAGRANULARITY + 1) + 0.5 * (WEEXTRADELAY - CSEXTRADELAY)) * GPMC_FCLK period

G = CYCLE2CYCLEDELAY * GPMC_FCLK period

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

H = ((OEOFFTIME - ACCESSTIME) * (TIMEPARAGRANULARITY + 1) + 0.5 * OEEXTRADELAY) * GPMC_FCLK period

I = ((OEOFFTIME + (N - 1) * PAGEBURSTACCESSTIME - CSONTIME) * (TIMEPARAGRANULARITY + 1) + 0.5 * (OEEXTRADELAY - CSEXTRADELAY)) * GPMC_FCLK period, where N = page burst access number

J = (CSONTIME * (TIMEPARAGRANULARITY + 1) + 0.5 * CSEXTRADELAY) * GPMC_FCLK period

K = ((ADVONTIME - CSONTIME) * (TIMEPARAGRANULARITY + 1) + 0.5 * (ADVEXTRADELAY - CSEXTRADELAY)) * GPMC_FCLK period

L = ((OEONTIME - CSONTIME) * (TIMEPARAGRANULARITY + 1) + 0.5 * (OEEXTRADELAY - CSEXTRADELAY)) * GPMC_FCLK period M = ((ACCESSTIME - CSONTIME) * (TIMEPARAGRANULARITY + 1) - 0.5 * CSEXTRADELAY) * GPMC_FCLK period

GPMC_$\overline{\text{BE0}}$_CLE/$\overline{\text{GPMC_BE1}}$ pulse:

For single read: N = RDCYCLETIME * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period

For burst read: N = (RDCYCLETIME + (N - 1) * PAGEBURSTACCESSTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period, where N = page burst access number

For burst write: N = (WRCYCLETIME + (N - 1) * PAGEBURSTACCESSTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period, where N = page burst access number

O = ((WRCYCLETIME + (N - 1) * PAGEBURSTACCESSTIME - CSONTIME) * (TIMEPARAGRANULARITY + 1) + 0.5 * (ADVEXTRADELAY - CSEXTRADELAY)) * GPMC_FCLK period

Figure 6-47 shows an asynchronous NOR single write simplified example where formulas are associated with signal waves.

Figure 6-47. Asynchronous NOR Single Write Simplified Example

Write multiple access is not supported in asynchronous mode. If WRITEMULTIPLE is enabled with WRITETYPE as asynchronous, the GPMC processes single asynchronous accesses.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.4 Use Cases

6.1.4.1 How to Set GPMC Timing Parameters for Typical Accesses

6.1.4.1.1 External Memory Attached to the GPMC Module

As discussed in the introduction to this chapter, the GPMC module supports the following external memory types:

- Asynchronous or synchronous, 8-bit or 16-bit-width memory or device
- 16-bit address/data-multiplexed or not multiplexed NOR flash device
- 8- or 16-bit NAND flash device

The following examples show how to calculate GPMC timing parameters by showing a typical parameter setup for the access to be performed.

The example is based on a 512-Mb multiplexed NOR flash memory with the following characteristics:

- Type: NOR flash (address/data-multiplexed mode)
- Size: 512M bits
- Data Bus: 16 bits wide
- Speed: 104 MHz clock frequency
- Read access time: 80 ns

6.1.4.1.2 Typical GPMC Setup

Table 6-49 lists some of the I/Os of the GPMC module.

Table 6-49. GPMC Signals

Signal Name	I/O	Description
GPMC_FCLK	Internal	Functional and interface clock. Acts as the time reference.
GPMC_CLK	O	External clock provided to the external device for synchronous operations
GPMC_A[27:17]	O	Address
GPMC_D[15: 0]	I/O	Data-multiplexed with addresses A[16:1] on memory side
$\overline{\text{GPMC_CSx}}$	O	Chip-select (where x = 0, or 1)
GPMC_$\overline{\text{ADV}}$_ALE	O	Address valid enable
$\overline{\text{GPMC_OE_RE}}$	O	Output enable (read access only)
$\overline{\text{GPMC_WE}}$	O	Write enable (write access only)
GPMC_WAIT[1:0]	I	Ready signal from memory device. Indicates when valid burst data is ready to be read

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 6-48 shows the typical connection between the GPMC module and an attached NOR Flash memory.

Figure 6-48. GPMC Connection to an External NOR Flash Memory

The following sections demonstrate how to calculate GPMC parameters for three access types:

- Synchronous burst read
- Asynchronous read
- Asynchronous single write

DRAFT

Copyright © 2011, Texas Instruments Incorporated

6.1.4.1.3 GPMC Configuration for Synchronous Burst Read Access

The clock runs at 104 MHz (f = 104 MHz; T = 9, 615 ns).

Table 6-50 shows the timing parameters (on the memory side) that determine the parameters on the GPMC side.

Table 6-51 shows how to calculate timings for the GPMC using the memory parameters.

Figure 6-49 shows the synchronous burst read access.

Table 6-50. Useful Timing Parameters on the Memory Side

AC Read Characteristics on the Memory Side	Description	Duration (ns)
tCES	$\overline{CS}$ setup time to clock	0
tACS	Address setup time to clock	3
tIACC	Synchronous access time	80
tBACC	Burst access time valid clock to output delay	5,2
tCEZ	Chip-select to High-Impedance	7
tOEZ	Output enable to High-Impedance	7
tAVC	$\overline{ADV}$ setup time	6
tAVD	$\overline{AVD}$ pulse	6
tACH	Address hold time from clock	3

The following terms, which describe the timing interface between the controller and its attached device, are used to calculate the timing parameters on the GPMC side:

- Read Access time (GPMC side): Time required to activate the clock + read access time requested on the memory side + data setup time required for optimal capture of a burst of data
- Data setup time (GPMC side): Ensures a good capture of a burst of data (as opposed to taking a burst of data out). One word of data is processed in one clock cycle (T = 9,615 ns). The read access time between 2 bursts of data is tBACC = 5,2 ns. Therefore, data setup time is a clock period - tBACC = 4,415 ns of data setup.
- Access completion (GPMC side): (Different from page burst access time) Time required between the last burst access and access completion: $\overline{CS}$/$\overline{OE}$ hold time ($\overline{CS}$ and $\overline{OE}$ must be released at the end of an access. These signals are held to allow the access to complete).
- Read cycle time (GPMC side): Read Access time + access completion
- Write cycle time for burst access: Not supported for NOR flash memory

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 6-51. Calculating GPMC Timing Parameters

Parameter Name on GPMC Side	Formula	Duration (ns)	Number of Clock Cycles (F = 104 MHz)	GPMC Register Configurations
GPMC FCLK Divider	-	-	-	GPMCFCLKDIVIDER = 0
ClkActivationTime	min (tCES, tACS)	3	1	CLKACTIVATIONTIME = 1
RdAccessTime	roundmax (ClkActivationTime + tIACC + DataSetupTime)	94,03: (9,615 + 80 + 4,415)	10 : roundmax (94,03 / 9,615)	ACCESSTIME = Ah
PageBurstAccessTime	roundmax (tBACC)	roundmax (5,2)	1	PAGEBURSTACCESSTIME = 1
RdCycleTime	AccessTime + max (tCEZ, tOEZ)	101, 03: (94, 03 + 7)	11	RDCYCLETIME = Bh
CsOnTime	tCES	0	0	CSONTIME = 0
CsReadOffTime	RdCycleTime	-	11	CSRDOFFTIME = Bh
AdvOnTime	tAVC	0	0	ADVONTIME = 0
AdvRdOffTime	tAVD + tAVC	12	2	ADVRDOFFTIME = 2h
OeOnTime	(ClkActivationTime + tACH) < OeOnTime < (ClkActivationTime + tIACC)	-	3, for instance	OEONTIME = 3h
OeOffTime	RdCycleTime	-	11	OEOFFTIME = Bh

Figure 6-49. Synchronous Burst Read Access (Timing Parameters in Clock Cycles)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.4.1.4 GPMC Configuration for Asynchronous Read Access

The clock runs at 104 MHz (f = 104 MHz; T = 9, 615 ns).

Table 6-52 shows the timing parameters (on the memory side) that determine the parameters on the GPMC side.

Table 6-53 shows how to calculate timings for the GPMC using the memory parameters.

Figure 6-50 shows the asynchronous read access.

Table 6-52. AC Characteristics for Asynchronous Read Access

AC Read Characteristics on the Memory Side	Description	Duration (ns)
tCE	Read Access time from $\overline{CS}$ low	80
tAAVDS	Address setup time to rising edge of $\overline{ADV}$	3
tAVDP	$\overline{ADV}$ low time	6
tCAS	$\overline{CS}$ setup time to $\overline{ADV}$	0
tOE	Output enable to output valid	6
tOEZ	Output enable to High-Impedance	7

Use the following formula to calculate the RdCycleTime parameter for this typical access:

RdCycleTime = RdAccessTime + AccessCompletion = RdAccessTime + 1 clock cycle + tOEZ

- First, on the memory side, the external memory makes the data available to the output bus. This is the memory-side read access time defined in Table 6-53: the number of clock cycles between the address capture ($\overline{ADV}$ rising edge) and the data valid on the output bus.

The GPMC requires some hold time to allow the data to be captured correctly and the access to be finished.

- To read the data correctly, the GPMC must be configure to meet the data setup time requirement of the memory; the GPMC module captures the data on the next rising edge. This is access time on the GPMC side.
- There must also be a data hold time for correctly reading the data (checking that there is no $\overline{OE}$/$\overline{CS}$ deassertion while reading the data). This data hold time is 1 clock cycle (that is, AccessTime + 1).
- To complete the access, $\overline{OE}$/$\overline{CS}$ signals are driven to high-impedance. AccessTime + 1 + tOEZ is the read cycle time.
- Addresses can now be relatched and a new read cycle begun.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 6-53. GPMC Timing Parameters for Asynchronous Read Access

Parameter Name on GPMC side	Formula	Duration (ns)	Number of Clock Cycles (F = 104 MHz)	GPMC Register Configurations
ClkActivationTime	n/a (asynchronous mode)			
AccessTime	round max (tCE)	80	9	ACCESSTIME = 9h
PageBurstAccessTime	n/a (single access)			
RdCycleTime	AccessTime + 1cycle + tOEZ	96, 615	11	RDCYCLETIME = Bh
CsOnTime	tCAS	0	0	CSONTIME = 0
CsReadOffTime	AccessTime + 1 cycle	89, 615	10	CSRDOFFTIME = Ah
AdvOnTime	tAAVDS	3	1	ADVONTIME = 1
AdvRdOffTime	tAAVDS + tAVDP	9	1	ADVRDOFFTIME = 1
OeOnTime	OeOnTime ≥ AdvRdOffTime (multiplexed mode)	-	3, for instance	OEONTIME = 3h
OeOffTime	AccessTime + 1cycle	89, 615	10	OEOFFTIME = Ah

Figure 6-50. Asynchronous Single Read Access (Timing Parameters in Clock Cycles)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.4.1.5 GPMC Configuration for Asynchronous Single Write Access

The clock runs at 104 MHz: (f = 104 MHz; T = 9, 615 ns).

Table 6-54 shows how to calculate timings for the GPMC using the memory parameters.

Table 6-55 shows the timing parameters (on the memory side) that determine the parameters on the GPMC side.

Figure 6-51 shows the synchronous burst write access.

Table 6-54. AC Characteristics for Asynchronous Single Write (Memory Side)

AC Characteristics on the Memory Side	Description	Duration (ns)
tWC	Write cycle time	60
tAVDP	$\overline{ADV}$ low time	6
tWP	Write pulse width	25
tWPH	Write pulse width high	20
tCS	$\overline{CS}$ setup time to $\overline{WE}$	3
tCAS	$\overline{CS}$ setup time to $\overline{ADV}$	0
tAVSC	$\overline{ADV}$ setup time	3

For asynchronous single write access, write cycle time is WrCycleTime = WeOffTime + AccessCompletion = WeOffTime + 1. For the AccesCompletion, the GPMC requires 1 cycle of data hold time ($\overline{CS}$ de-assertion).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 6-55. GPMC Timing Parameters for Asynchronous Single Write

Parameter Name on GPMC side	Formula	Duration (ns)	Number of Clock Cycles (F = 104 MHz)	GPMC Register Configurations
ClkActivationTime	n/a (asynchronous mode)			
AccessTime	Applicable only to WAITMONITORING (the value is the same as for read access)			
PageBurstAccessTime	n/a (single access)			
WrCycleTime	WeOffTime + AccessCompletion	57, 615	6	WRCYCLETIME = 6h
CsOnTime	tCAS	0	0	CSONTIME = 0
CsWrOffTime	WeOffTime + 1	57, 615	6	CSWROFFTIME = 6h
AdvOnTime	tAVSC	3	1	ADVONTIME = 1
AdvWrOffTime	tAVSC + tAVDP	9	1	ADVWROFFTIME = 1
WeOnTime	tCS	3	1	WEONTIME = 1
WeOffTime	tCS + tWP + tWPH	48	5	WEOFFTIME = 5h

Figure 6-51. Asynchronous Single Write Access (Timing Parameters in Clock Cycles)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.4.2 How to Choose a Suitable Memory to Use With the GPMC

This section is intended to help the user select a suitable memory device to interface with the GPMC controller.

6.1.4.2.1 Supported Memories or Devices

NAND flash and NOR flash architectures are the two flash technologies. The GPMC supports various types of external memory or device, basically any one that supports NAND or NOR protocols:

- 8- and 16-bit width asynchronous or synchronous memory or device (8-bit: non burst device only)
- 16-bit address and data multiplexed NOR flash devices (pSRAM, , ...)
- 8- and 16-bit NAND flash device

6.1.4.2.2 Memory Pin Multiplexing

This section highlights the interfacing differences of the GPMC supported memories (see Table 6-56).

Table 6-56. Supported Memory Interfaces

Function	16-bit Address/Data Non-Muxed Flash	16-bit Address/Data Muxed pSRAM or NOR Flash	OneNAND	16-bit NAND	8-bit NAND
GPMC_A[27]	A28	A28			
GPMC_A[26]	A27	A27			
GPMC_A[25]	A26	A26			
GPMC_A[24]	A25	A25			
GPMC_A[23]	A24	A24			
GPMC_A[22]	A23	A23			
GPMC_A[21]	A22	A22			
GPMC_A[20]	A21	A21			
GPMC_A[19]	A20	A20			
GPMC_A[18]	A10	A10			
GPMC_A[17]	A18	A18			
GPMC_A[16]	A17	A17			
GPMC_A[15]	A16				
GPMC_A[14]	A15				
GPMC_A[13]	A14				
GPMC_A[12]	A13				
GPMC_A[11]	A12				
GPMC_A[10]	A11				
GPMC_A[9]	A10				
GPMC_A[8]	A9				
GPMC_A[7]	A8				
GPMC_A[6]	A7				
GPMC_A[5]	A6				
GPMC_A[4]	A5				
GPMC_A[3]	A4				
GPMC_A[2]	A3				
GPMC_A[1]	A2				
GPMC_A[0]	A1				
GPMC_D[15]	D15 or A16	D15 or A16	D15 or A16	IO15	
GPMC_D[14]	D14 or A15	D14 or A15	D14 or A15	IO14	
GPMC_D[13]	D13 or A14	D13 or A14	D13 or A14	IO13	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 6-56. Supported Memory Interfaces (continued)

Function	16-bit Address/Data Non-Muxed Flash	16-bit Address/Data Muxed pSRAM or NOR Flash	OneNAND	16-bit NAND	8-bit NAND
GPMC_D[12]	D12 or A13	D12 or A13	D12 or A13	IO12	
GPMC_D[11]	D11 or A12	D11 or A12	D11 or A12	IO11	
GPMC_D[10]	D10 or A11	D10 or A11	D10 or A11	IO10	
GPMC_D[9]	D9 or A10	D9 or A10	D9 or A10	IO9	
GPMC_D[8]	D8 or A9	D8 or A9	D8 or A9	IO8	
GPMC_D[7]	D7 or A8	D7 or A8	D7 or A8	IO7	IO7
GPMC_D[6]	D6 or A7	D6 or A7	D6 or A7	IO6	IO6
GPMC_D[5]	D5 or A6	D5 or A6	D5 or A6	IO5	IO5
GPMC_D[4]	D4 or A5	D4 or A5	D4 or A5	IO4	IO4
GPMC_D[3]	D3 or A4	D3 or A4	D3 or A4	IO3	IO3
GPMC_D[2]	D2 or A3	D2 or A3	D2 or A3	IO2	IO2
GPMC_D[1]	D1 or A2	D1 or A2	D1 or A2	IO1	IO1
GPMC_D[0]	D0 or A1	D0 or A1	D0 or A1	IO0	IO0
GPMC_CLK	CLK	CLK	CLK		
$\overline{\text{GPMC_CS[0]}}$	$\overline{\text{CS0}}$ (Chip Select)	$\overline{\text{CS0}}$ (Chip Select)	$\overline{\text{CS0}}$ (Chip Select)	$\overline{\text{CE0}}$ (Chip Enable)	$\overline{\text{CE0}}$ (Chip Enable)
$\overline{\text{GPMC_CS[1]}}$	$\overline{\text{CS1}}$	$\overline{\text{CS1}}$	$\overline{\text{CS1}}$	$\overline{\text{CE1}}$	$\overline{\text{CE1}}$
$\overline{\text{GPMC_CS[2]}}$	$\overline{\text{CS2}}$	$\overline{\text{CS2}}$	$\overline{\text{CS2}}$	$\overline{\text{CE2}}$	$\overline{\text{CE2}}$
$\overline{\text{GPMC_CS[3]}}$	$\overline{\text{CS3}}$	$\overline{\text{CS3}}$	$\overline{\text{CS3}}$	$\overline{\text{CE3}}$	$\overline{\text{CE3}}$
$\overline{\text{GPMC_CS[4]}}$	$\overline{\text{CS4}}$	$\overline{\text{CS4}}$	$\overline{\text{CS4}}$	$\overline{\text{CE4}}$	$\overline{\text{CE4}}$
$\overline{\text{GPMC_CS[5]}}$	$\overline{\text{CS5}}$	$\overline{\text{CS5}}$	$\overline{\text{CS5}}$	$\overline{\text{CE5}}$	$\overline{\text{CE5}}$
$\overline{\text{GPMC_CS[6]}}$	$\overline{\text{CS6}}$	$\overline{\text{CS6}}$	$\overline{\text{CS6}}$	$\overline{\text{CE6}}$	$\overline{\text{CE6}}$
$\overline{\text{GPMC_CS[7]}}$	$\overline{\text{CS7}}$	$\overline{\text{CS7}}$	$\overline{\text{CS7}}$	$\overline{\text{CE7}}$	$\overline{\text{CE7}}$
GPMC_$\overline{\text{ADV}}$_ALE	$\overline{\text{ADV}}$ (Address Value)	$\overline{\text{ADV}}$ (Address Value)	$\overline{\text{ADV}}$ (Address Value)	ALE (address latch enable)	ALE (address latch enable)
$\overline{\text{GPMC_OE_RE}}$	$\overline{\text{OE}}$ (Output Enable)	$\overline{\text{OE}}$ (Output Enable)	$\overline{\text{OE}}$ (Output Enable)	$\overline{\text{RE}}$ (read enable)	$\overline{\text{RE}}$ (read enable)
$\overline{\text{GPMC_WE}}$	$\overline{\text{WE}}$ (Write Enable)	$\overline{\text{WE}}$ (Write Enable)	$\overline{\text{WE}}$ (Write Enable)	$\overline{\text{WE}}$ (write enable)	$\overline{\text{WE}}$ (write enable)
GPMC_$\overline{\text{BE0}}$_CLE	$\overline{\text{BE0}}$ (Byte Enable)	$\overline{\text{BE0}}$ (Byte Enable)	$\overline{\text{BE0}}$ (Byte Enable)	CLE (command latch enable)	CLE (command latch enable)
$\overline{\text{GPMC_BE1}}$	$\overline{\text{BE1}}$	$\overline{\text{BE1}}$	$\overline{\text{BE1}}$		
$\overline{\text{GPMC_WP}}$	$\overline{\text{WP}}$ (Write Protect)	$\overline{\text{WP}}$ (Write Protect)	$\overline{\text{WP}}$ (Write Protect)	$\overline{\text{WP}}$ (write protect)	$\overline{\text{WP}}$ (write protect)
GPMC_WAIT0	WAIT0	WAIT0	WAIT0	R/$\overline{\text{B0}}$ (ready/busy)	R/$\overline{\text{B0}}$ (ready/busy)
GPMC_WAIT1	WAIT1	WAIT1	WAIT1	R/$\overline{\text{B1}}$ (ready/busy)	R/$\overline{\text{B1}}$ (ready/busy)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.4.2.3 NAND Interface Protocol

NAND flash architecture, introduced in 1989, is a flash technology. NAND is a page-oriented memory device, that is, read and write accesses are done by pages. NAND achieves great density by sharing common areas of the storage transistor, which creates strings of serially connected transistors (in NOR devices, each transistor stands alone). Thanks to its high density NAND is best suited to devices requiring high capacity data storage, such as pictures, music, or data files. NAND non-volatility, makes of it a good storage solution for many applications where mobility, low power, and speed are key factors. Low pin count and simple interface are other advantages of NAND.

Table 6-57 summarizes the NAND interface signals level applied to external device or memories.

Table 6-57. NAND Interface Bus Operations Summary

Bus Operation	CLE	ALE	$\overline{CE}$	$\overline{WE}$	$\overline{RE}$	$\overline{WP}$
Read (cmd input)	H	L	L	RE	H	x
Read (add input)	L	H	L	RE	H	x
Write (cmd input)	H	L	L	RE	H	H
Write (add input)	L	H	L	RE	H	H
Data input	L	L	L	RE	H	H
Data output	L	L	L	H	FE	x
Busy (during read)	x	x	H	H	H	x
Busy (during program)	x	x	x	x	x	H
Busy (during erase)	x	x	x	x	x	H
Write protect	x	x	x	x	x	L
Stand-by	x	x	H	x	x	H/L

6.1.4.2.4 NOR Interface Protocol

NOR flash architecture, introduced in 1988, is a flash technology. Unlike NAND, which is a sequential access device, NOR is directly addressable; i.e., it is designed to be a random access device. NOR is best suited to devices used to store and run code or firmware, usually in small capacities. While NOR has fast read capabilities it has slow write and erase functions compared to NAND architecture.

Table 6-58 summarizes the NOR interface signals level applied to external device or memories.

Table 6-58. NOR Interface Bus Operations Summary

Bus Operation	CLK	$\overline{ADV}$	$\overline{CS}$	$\overline{OE}$	$\overline{WE}$	WAIT	DQ[15:0]
Read (asynchronous)	x	L	L	L	H	Asserted	Output
Read (synchronous)	Running	L	L	L	H	Driven	Output
Read (burst suspend)	Halted	x	L	H	H	Active	Output
Write	x	L	L	H	L	Asserted	Input
Output disable	x	x	L	H	H	Asserted	High-Z
Standby	x	x	H	x	x	High-Z	High-Z

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.4.2.5 Other Technologies

Other supported device type interact with the GPMC through the NOR interface protocol.

OneNAND Flash is a high-density and low-power memory device. It is based on single- or multi-level-cell NAND core with SRAM and logic, and interfaces as a synchronous NOR Flash, plus has synchronous write capability. It reads faster than conventional NAND and writes faster than conventionnal NOR flash. Hence, it is appropriate for both mass storage and code storage.

pSRAM stands for pseudo-static random access memory. pSRAM is a low-power memory device for mobile applications. pSRAM is based on the DRAM cell with internal refresh and address control features, and interfaces as a synchronous NOR Flash, plus has synchronous write capability.

6.1.4.2.6 Supported Protocols

The GPMC supports the following interface protocols when communicating with external memory or external devices:

- Asynchronous read/write access
- Asynchronous read page access (4-8-16 Word16)
- Synchronous read/write access
- Synchronous read burst access without wrap capability (4-8-16 Word16)
- Synchronous read burst access with wrap capability (4-8-16 Word16)

6.1.4.2.7 GPMC Features and Settings

This section lists GPMC features and settings:

- Supported device type: up to four NAND or NOR protocol external memories or devices
- Operating Voltage: 3.3V
- Maximum GPMC addressing capability: 512 MBytes divided into eight chip-selects
- Maximum supported memory size: 256 MBytes (must be a power-of-2)
- Minimum supported memory size: 16 MBytes (must be a power-of-2). Aliasing occurs when addressing smaller memories.
- Data path to external memory or device: 8- and 16-bit wide
- Burst and page access: burst of 4-8-16 Word16
- Supports bus keeping
- Supports bus turn around

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.5 GPMC Registers

Table 6-59 provides a summary of the GPMC registers. All GPMC registers are aligned to 32-bit address boundaries. All register file accesses, except the GPMC_NAND_DATA_i register, are little endian. If the GPMC_NAND_DATA_i register is accessed, the endianness is access-dependent.

In this section, i corresponds to the chip-select number, i = 0 to 7.

Table 6-59. GPMC Registers

Address Offset	Register Name	Section
0h	GPMC_REVISION	Section 6.1.5.1
10h	GPMC_SYSCONFIG	Section 6.1.5.2
14h	GPMC_SYSSTATUS	Section 6.1.5.3
18h	GPMC_IRQSTATUS	Section 6.1.5.4
1Ch	GPMC_IRQENABLE	Section 6.1.5.5
40h	GPMC_TIMEOUT_CONTROL	Section 6.1.5.6
44h	GPMC_ERR_ADDRESS	Section 6.1.5.7
48h	GPMC_ERR_TYPE	Section 6.1.5.8
50h	GPMC_CONFIG	Section 6.1.5.9
54h	GPMC_STATUS	Section 6.1.5.10
60h + (30h × i)	GPMC_CONFIG1_i[(1)]	Section 6.1.5.11
64h + (30h × i)	GPMC_CONFIG2_i[(1)]	Section 6.1.5.12
68h + (30h × i)	GPMC_CONFIG3_i[(1)]	Section 6.1.5.13
6Ch + (30h × i)	GPMC_CONFIG4_i[(1)]	Section 6.1.5.14
70h + (30h × i)	GPMC_CONFIG5_i[(1)]	Section 6.1.5.15
74h + (30h × i)	GPMC_CONFIG6_i[(1)]	Section 6.1.5.16
78h + (30h × i)	GPMC_CONFIG7_i[(1)]	Section 6.1.5.17
7Ch + (30h × i)	GPMC_NAND_COMMAND_i[(1)]	Section 6.1.5.18
80h + (30h × i)	GPMC_NAND_ADDRESS_i[(1)]	Section 6.1.5.19
84h + (30h × i)	GPMC_NAND_DATA_i[(1)]	Section 6.1.5.20
1E0h	GPMC_PREFETCH_CONFIG1	Section 6.1.5.21
1E4h	GPMC_PREFETCH_CONFIG2	Section 6.1.5.22
1ECh	GPMC_PREFETCH_CONTROL	Section 6.1.5.23
1F0h	GPMC_PREFETCH_STATUS	Section 6.1.5.24
1F4h	GPMC_ECC_CONFIG	Section 6.1.5.25
1F8h	GPMC_ECC_CONTROL	Section 6.1.5.26
1FCh	GPMC_ECC_SIZE_CONFIG	Section 6.1.5.27
200h + (4h × j)	GPMC_ECCj_RESULT[(2)]	Section 6.1.5.28
240h + (10h × i)	GPMC_BCH_RESULT0_i[(1)]	Section 6.1.5.29
244h + (10h × i)	GPMC_BCH_RESULT1_i[(1)]	Section 6.1.5.30
248h + (10h × i)	GPMC_BCH_RESULT2_i[(1)]	Section 6.1.5.31
24Ch + (10h × i)	GPMC_BCH_RESULT3_i[(1)]	Section 6.1.5.32
300h + (10h × i)	GPMC_BCH_RESULT4_i[(1)]	Section 6.1.5.34
304h + (10h × i)	GPMC_BCH_RESULT5_i[(1)]	Section 6.1.5.35
308h + (10h × i)	GPMC_BCH_RESULT6_i[(1)]	Section 6.1.5.36
2D0h	GPMC_BCH_SWDATA	Section 6.1.5.33

(1) i = 0 to 7 for GPMC
(2) j = 0 to 8 for GPMC

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.5.1 GPMC_REVISION

This register contains the IP revision code.

Figure 6-52. GPMC_REVISION

31–16	15–8	7–0
Reserved	Reserved	REV
R-0	R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 6-60. GPMC_REVISION Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-0	REV	0-FFh	IP revision. Major revision is [7-4]. Minor revision is [3-0]. Examples: 10h for revision 1.0, 21h for revision 2.1.

6.1.5.2 GPMC_SYSCONFIG

This register controls the various parameters of the OCP interface.

Figure 6-53. GPMC_SYSCONFIG

31–16	15–5	4–3	2	1	0
Reserved	Reserved	SIDLEMODE	Rsvd	SOFTRESET	AUTOIDLE
R-0	R-0	R/W	R-0	R/W	R/W

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 6-61. GPMC_SYSCONFIG Field Descriptions

Bit	Field	Value	Description
31-5	Reserved	0	Reserved
4-3	SIDLEMODE		Idle mode
		0	Force-idle. An idle request is acknowledged unconditionally
		1h	No-idle. An idle request is never acknowledged
		2h	Smart-idle. Acknowledgement to an idle request is given based on the internal activity of the module
		3h	Reserved
2	Reserved	0	Reserved
1	SOFTRESET		Software reset (Set 1 to this bit triggers a module reset. This bit is automatically reset by hardware. During reads, it always returns 0)
		0	Normal mode
		1	The module is reset
0	AUTOIDLE		Internal OCP clock gating strategy
		0	Interface clock is free-running
		1	Automatic Interface clock gating strategy is applied, based on the Interconnect activity

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.5.3 GPMC_SYSSTATUS

This register provides status information about the module, excluding the interrupt status information.

Figure 6-54. GPMC_SYSSTATUS

31		16
Reserved		
R-0		

15	1	0
Reserved		RESETDONE
R-0		R-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 6-62. GPMC_SYSSTATUS Field Descriptions

Bit	Field	Value	Description
31-1	Reserved	0	Reserved
0	RESETDONE		Internal reset monitoring
		R0	Internal module reset in on-going
		R1	Reset completed

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.5.4 GPMC_IRQSTATUS

This interrupt status register regroups all the status of the module internal events that can generate an interrupt.

Figure 6-55. GPMC_IRQSTATUS

31			16
Reserved			
R-0			

15	10	9	8
Reserved		WAIT1EDGE DETECTIONSTATUS	WAIT0EDGE DETECTIONSTATUS
R-0		R/W-0	R/W-0

7	2	1	0
Reserved		TERMINAL COUNTSTATUS	FIFOEVENT STATUS
R-0		R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 6-63. GPMC_IRQSTATUS Field Descriptions

Bit	Field	Value	Description
31-10	Reserved	0	Reserved
9	WAIT1EDGEDETECTIONSTATUS		Status of the Wait1 Edge Detection interrupt
		R0	A transition on WAIT1 input pin has not been detected
		W0	WAIT1EDGEDETECTIONSTATUS bit unchanged
		R1	A transition on WAIT1 input pin has been detected
		W1	WAIT1EDGEDETECTIONSTATUS bit is reset
8	WAIT0EDGEDETECTIONSTATUS		Status of the Wait0 Edge Detection interrupt
		R0	A transition on WAIT0 input pin has not been detected
		W0	WAIT0EDGEDETECTIONSTATUS bit unchanged
		R1	A transition on WAIT0 input pin has been detected
		W1	WAIT0EDGEDETECTIONSTATUS bit is reset
7-2	Reserved	0	Reserved
1	TERMINALCOUNTSTATUS		Status of the TerminalCountEvent interrupt
		R0	Indicates that CountValue is greater than 0
		W0	TERMINALCOUNTSTATUS bit unchanged
		R1	Indicates that CountValue is equal to 0
		W1	TERMINALCOUNTSTATUS bit is reset
0	FIFOEVENTSTATUS		Status of the FIFOEvent interrupt
		R0	Indicates than less than GPMC_PREFETCH_STATUS[16] FIFOTHRESHOLDSTATUS bytes are available in prefetch mode and less than FIFOTHRESHOLD bytes free places are available in write-posting mode.
		W0	FIFOEVENTSTATUS bit unchanged
		R1	Indicates than at least GPMC_PREFETCH_STATUS[16] FIFOTHRESHOLDSTATUS bytes are available in prefetch mode and at least FIFOTHRESHOLD bytes free places are available in write-posting mode.
		W1	FIFOEVENTSTATUS bit is reset

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.5.5 GPMC_IRQENABLE

The interrupt enable register allows to mask/unmask the module internal sources of interrupt, on a event-by-event basis.

Figure 6-56. GPMC_IRQENABLE

31			16
Reserved			
R-0			

15	10	9	8
Reserved		WAIT1EDGE DETECTIONENABLE	WAIT0EDGE DETECTIONENABLE
R-0		R/W-0	R/W-0

7	2	1	0
Reserved		TERMINAL COUNTENABLE	FIFOEVENT ENABLE
R-0		R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 6-64. GPMC_IRQENABLE Field Descriptions

Bit	Field	Value	Description
31-10	Reserved	0	Reserved
9	WAIT1EDGEDETECTIONENABLE		Enables the Wait1 Edge Detection interrupt
		0	Wait1EdgeDetection interrupt is masked
		1	Wait1EdgeDetection event generates an interrupt if occurs
8	WAIT0EDGEDETECTIONENABLE		Enables the Wait0 Edge Detection interrupt
		0	Wait0EdgeDetection interrupt is masked
		1	Wait0EdgeDetection event generates an interrupt if occurs
7-2	Reserved	0	Reserved
1	TERMINALCOUNTEVENTENABLE		Enables TerminalCountEvent interrupt issuing in pre-fetch or write posting mode
		0	TerminalCountEvent interrupt is masked
		1	TerminalCountEvent interrupt is not masked
0	FIFOEVENTENABLE		Enables the FIFOEvent interrupt
		0	FIFOEvent interrupt is masked
		1	FIFOEvent interrupt is not masked

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.5.6 GPMC_TIMEOUT_CONTROL

The GPMC_TIMEOUT_CONTROL register allows the user to set the start value of the timeout counter

Figure 6-57. GPMC_TIMEOUT_CONTROL

31–16	15–13	12–4	3–1	0
Reserved	Reserved	TIMEOUTSTARTVALUE	Reserved	TIMEOUTENABLE
R-0	R-0	R/W-0	R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 6-65. GPMC_TIMEOUT_CONTROL Field Descriptions

Bit	Field	Value	Description
31-13	Reserved	0	Reserved
12-4	TIMEOUTSTARTVALUE	0-1FFh	Start value of the time-out counter (000 corresponds to 0 GPMC.FCLK cycle, 1h corresponds to 1 GPMC.FCLK cycle, and 1FFh corresponds to 511 GPMC.FCLK cycles)
3-1	Reserved	0	Reserved
0	TIMEOUTENABLE		Enable bit of the TimeOut feature
		0	TimeOut feature is disabled
		1	TimeOut feature is enabled

6.1.5.7 GPMC_ERR_ADDRESS

The GPMC_ERR_ADDRESS register stores the address of the illegal access when an error occurs.

Figure 6-58. GPMC_ERR_ADDRESS

31	30–0
Rsvd	ILLEGALADD
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 6-66. GPMC_ERR_ADDRESS Field Descriptions

Bit	Field	Value	Description
31	Reserved	0	Reserved
30-0	ILLEGALADD	0-7FFF FFFFh	Address of illegal access: A30 (0 for memory region, 1 for GPMC register region) and A29-A0 (1GByte maximum)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.5.8 GPMC_ERR_TYPE

The GPMC_ERR_TYPE register stores the type of error when an error occurs.

Figure 6-59. GPMC_ERR_TYPE

31–16	15–11	10–8	7–5	4	3	2	1	0
Reserved	Reserved	ILLEGALMCMD	Reserved	ERRORNOT SUPPADD	ERRORNOT SUPPMCMD	ERROR TIMEOUT	Reserved	ERRORVALID
R-0	R-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 6-67. GPMC_ERR_TYPE Field Descriptions

Bit	Field	Value	Description
31-11	Reserved	0	Reserved
10-8	ILLEGALMCMD	0-7h	System Command of the transaction that caused the error
7-5	Reserved	0	Reserved
4	ERRORNOTSUPPADD		Not supported Address error
		0	No error occurs
		1	The error is due to a non supported Address
3	ERRORNOTSUPPMCMD		Not supported Command error
		0	No error occurs
		1	The error is due to a non supported Command
2	ERRORTIMEOUT		Time-out error
		0	No error occurs
		1	The error is due to a time out
1	Reserved	0	Reserved
0	ERRORVALID		Error validity status - Must be explicitly cleared with a write 1 transaction
		0	All error fields no longer valid
		1	Error detected and logged in the other error fields

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.5.9 GPMC_CONFIG

The configuration register allows global configuration of the GPMC.

Figure 6-60. GPMC_CONFIG

31–16	15–10	9	8	7–5	4	3–2	1	0
Reserved	Reserved	WAIT1PIN POLARITY	WAIT0PIN POLARITY	Reserved	WRITE PROTECT	Reserved	LIMITED ADDRESS	NANDFORCE POSTEDWRITE
R-0	R-0	R/W-0	R/W-0	R-0	R/W-0	R-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 6-68. GPMC_CONFIG Field Descriptions

Bit	Field	Value	Description
31-10	Reserved	0	Reserved
9	WAIT1PINPOLARITY		Selects the polarity of input pin WAIT1
		0	WAIT1 active low
		1	WAIT1 active high
8	WAIT0PINPOLARITY		Selects the polarity of input pin WAIT0
		0	WAIT0 active low
		1	WAIT0 active high
7-5	Reserved	0	Reserved
4	WRITEPROTECT		Controls the $\overline{WP}$ output pin level
		0	$\overline{WP}$ output pin is low
		1	$\overline{WP}$ output pin is high
3-2	Reserved	0	Reserved
1	LIMITEDADDRESS	0-1	Limited Address device support
0	NANDFORCEPOSTEDWRITE		Enables the Force Posted Write feature to NAND Cmd/Add/Data location
		0	Disables Force Posted Write
		1	Enables Force Posted Write

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.5.10 GPMC_STATUS

The status register provides global status bits of the GPMC.

Figure 6-61. GPMC_STATUS

31			16
Reserved			
R-0			

15	10	9	8
Reserved		WAIT1STATUS	WAIT0STATUS
R-0		R/W-0	R/W-0

7	1	0
Reserved		EMPTYWRITEBUFFERSTATUS
R-0		R/W-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 6-69. GPMC_STATUS Field Descriptions

Bit	Field	Value	Description
31-10	Reserved	0	Reserved
9	WAIT1STATUS		Is a copy of input pin WAIT1. (Reset value is WAIT1 input pin sampled at IC reset)
		0	WAIT1 asserted (inactive state)
		1	WAIT1 de-asserted
8	WAIT0STATUS		Is a copy of input pin WAIT0. (Reset value is WAIT0 input pin sampled at IC reset)
		0	WAIT0 asserted (inactive state)
		1	WAIT0 de-asserted
7-1	Reserved	0	Reserved
0	EMPTYWRITEBUFFERSTATUS		Stores the empty status of the write buffer
		0	Write Buffer is not empty
		1	Write Buffer is empty

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.5.11 GPMC_CONFIG1_i

The configuration 1 register sets signal control parameters per chip select.

Figure 6-62. GPMC_CONFIG1_i

31	30	29	28	27	26	25	24
WRAPBURST	READMULTIPLE	READTYPE	WRITEMULTIPLE	WRITETYPE	CLKACTIVATIONTIME		ATTACHEDDEVICE PAGELENGTH
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		R/W-0

23	22	21	20	19	18	17	16
ATTACHEDDEVICE PAGELENGTH	WAITREADMONITORING	WAITWRITEMONITORING	Reserved	WAITMONITORINGTIME		WAITPINSELECT	
R/W-0	R/W-0	R/W-0	R-0	R/W-0		R/W-0	

15	14	13	12	11	10	9	8
Reserved		DEVICESIZE		DEVICETYPE		MUXADDDATA	
R-0		R/W-0		R/W-0		R/W-0	

7	5	4	3	2	1	0
Reserved		TIMEPARA GRANULARITY	Reserved		GPMCFCLKDIVIDER	
R-0		R/W-0	R-0		R/W-0	

LEGEND: R = Read only; W1C = Write 1 to clear bit; -*n* = value after reset

Table 6-70. GPMC_CONFIG1_i Field Descriptions

Bit	Field	Value	Description
31	WRAPBURST		Enables the wrapping burst capability. Must be set if the attached device is configured in wrapping burst
		0	Synchronous wrapping burst not supported
		1	Synchronous wrapping burst supported
30	READMULTIPLE		Selects the read single or multiple access
		0	single access
		1	multiple access (burst if synchonous, page if asynchronous)
29	READTYPE		Selects the read mode operation
		0	Read Asynchronous
		1	Read Synchronous
28	WRITEMULTIPLE		Selects the write single or multiple access
		0	Single access
		1	Multiple access (burst if synchronous, considered as single if asynchronous)
27	WRITETYPE		Selects the write mode operation
		0	Write Asynchronous
		1	Write Synchronous
26-25	CLKACTIVATIONTIME		Output GPMC.CLK activation time
		0	First rising edge of GPMC_CLK at start access time
		1h	First rising edge of GPMC_CLK one GPMC_FCLK cycle after start access time
		2h	First rising edge of GPMC_CLK two GPMC_FCLK cycles after start access time
		3h	Reserved
24-23	ATTACHEDDEVICEPAGELENGTH		Specifies the attached device page (burst) length (1 Word = Interface size)
		0	4 Words
		1h	8 Words
		2h	16 Words
		3h	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 6-70. GPMC_CONFIG1_i Field Descriptions (continued)

Bit	Field	Value	Description
22	WAITREADMONITORING		Selects the Wait monitoring configuration for Read accesses (Reset value is BOOTWAITEN input pin sampled at IC reset)
		0	WAIT pin is not monitored for read accesses
		1	WAIT pin is monitored for read accesses
21	WAITWRITEMONITORING		Selects the Wait monitoring configuration for Write accesses
		0	WAIT pin is not monitored for write accesses
		1	WAIT pin is monitored for write accesses
20	Reserved	0	Reserved
19-18	WAITMONITORINGTIME		Selects input pin Wait monitoring time
		0	WAIT pin is monitored with valid data
		1h	WAIT pin is monitored one GPMC_CLK cycle before valid data
		2h	WAIt pin is monitored two GPMC_CLK cycle before valid data
		3h	Reserved
17-16	WAITPINSELECT		Selects the input WAIT pin for this chip select (Reset value is BOOTWAITSELECT input pin sampled at IC reset for CS0 and 0 for CS1-7)
		0	WAIT input pin is WAIT0
		1h	WAIT input pin is WAIT1
		2h	Reserved
		3h	Reserved
15-14	Reserved	0	Reserved
13-12	DEVICESIZE		Selects the device size attached (Reset value is BOOTDEVICESIZE input pin sampled at IC reset for $\overline{CS}$[0] and 01 for $\overline{CS}$[1-7])
		0	8 bit
		1h	16 bit
		2h	Reserved
		3h	Reserved
11-10	DEVICETYPE		Selects the attached device type
		0	NOR Flash like, asynchronous and synchronous devices
		1h	Reserved
		2h	NAND Flash like devices, stream mode
		3h	Reserved
9-8	MUXADDDATA		Enables the Address and data multiplexed protocol (Reset value is CS0MUXDEVICE input pin sampled at IC reset for $\overline{CS}$[0] and 0 for $\overline{CS}$[1-7])
		0	Non-multiplexed attached device
		1h	AAD-multiplexed protocol device
		2h	Address and data multiplexed attached device
		3h	Reserved
7-5	Reserved	0	Reserved
4	TIMEPARAGRANULARITY		Signals timing latencies scalar factor (Rd/WRCycleTime, AccessTime, PageBurstAccessTime, CSOnTime, CSRd/WrOffTime, ADVOnTime, ADVRd/WrOffTime, OEOnTime, OEOffTime, WEOnTime, WEOffTime, Cycle2CycleDelay, BusTurnAround, TimeOutStartValue)
		0	×1 latencies
		1	×2 latencies
3-2	Reserved	0	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 6-70. GPMC_CONFIG1_i Field Descriptions (continued)

Bit	Field	Value	Description
1-0	GPMCFCLKDIVIDER		Divides the GPMC.FCLK clock
		0	GPMC_CLK frequency = GPMC_FCLK frequency
		1h	GPMC_CLK frequency = GPMC_FCLK frequency/2
		2h	GPMC_CLK frequency = GPMC_FCLK frequency/3
		3h	GPMC_CLK frequency = GPMC_FCLK frequency/4

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.5.12 GPMC_CONFIG2_i

Chip-select signal timing parameter configuration.

Figure 6-63. GPMC_CONFIG2_i

31–21	20–16
Reserved	CSWROFFTIME
R-0	R/W-0

15–13	12–8	7	6–4	3–0
Reserved	CSRDOFFTIME	CSEXTRADELAY	Reserved	CSONTIME
R-0	R/W-0	R/W-0	R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 6-71. GPMC_CONFIG2_i Field Descriptions

Bit	Field	Value	Description
31-21	Reserved	0	Reserved
20-16	CSWROFFTIME		CS# de-assertion time from start cycle time for write accesses
		0	0 GPMC_FCLK cycle
		1h	1 GPMC_FCLK cycle
		⋮	⋮
		1Fh	31 GPMC_FCLK cycles
15-13	Reserved	0	Reserved
12-8	CSRDOFFTIME		CS# de-assertion time from start cycle time for read accesses
		0	0 GPMC_FCLK cycle
		1h	1 GPMC_FCLK cycle
		⋮	⋮
		1Fh	31 GPMC_FCLK cycles
7	CSEXTRADELAY		CS# Add Extra Half GPMC.FCLK cycle
		0	CS i Timing control signal is not delayed
		1	CS i Timing control signal is delayed of half GPMC_FCLK clock cycle
6-4	Reserved	0	Reserved
3-0	CSONTIME		CS# assertion time from start cycle time
		0	0 GPMC_FCLK cycle
		1h	1 GPMC_FCLK cycle
		⋮	⋮
		1Fh	15 GPMC_FCLK cycles

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.5.13 GPMC_CONFIG3_i

ADV# signal timing parameter configuration.

Figure 6-64. GPMC_CONFIG3_i

31	30 – 28	27	26 – 24
Reserved	ADVAADMUXWROFFTIME	Reserved	ADVAADMUXRDOFFTIME
R-0	R/W-0	R-0	R/W-0

23 – 21	20 – 16
Reserved	ADVWROFFTIME
R/W-0	R/W-0

15 – 13	12 – 8
Reserved	ADVRDOFFTIME
R-0	R/W-0

7	6 – 4	3 – 0
ADVEXTRA DELAY	ADVAADMUXONTIME	ADVONTIME
R/W-0	R/W-0	R/W-0

LEGEND: R = Read only; W1C = Write 1 to clear bit; -*n* = value after reset

Table 6-72. GPMC_CONFIG3_i Field Descriptions

Bit	Field	Value	Description
31	Reserved	0	Reserved
30-28	ADVAADMUXWROFFTIME		ADV# de-assertion for first address phase when using the AAD-Mux protocol
		0	0 GPMC_FCLK cycle
		1h	1 GPMC_FCLK cycle
		⋮	⋮
		7h	7 GPMC_FCLK cycles
27	Reserved	0	Reserved
26-24	ADVAADMUXRDOFFTIME		ADV# assertion for first address phase when using the AAD-Mux protocol
		0	0 GPMC_FCLK cycle
		1h	1 GPMC_FCLK cycle
		⋮	⋮
		7h	7 GPMC_FCLK cycles
23-21	Reserved	0	Reserved
20-16	ADVWROFFTIME		ADV# de-assertion time from start cycle time for write accesses
		0	0 GPMC_FCLK cycle
		1h	1 GPMC_FCLK cycle
		⋮	⋮
		1Fh	31 GPMC_FCLK cycles
15-13	Reserved	0	Reserved
12-8	ADVRDOFFTIME		ADV# de-assertion time from start cycle time for read accesses
		0	0 GPMC_FCLK cycle
		1h	1 GPMC_FCLK cycle
		⋮	⋮
		1Fh	31 GPMC_FCLK cycles
7	ADVEXTRADELAY		ADV# Add Extra Half GPMC.FCLK cycle
		0	$\overline{ADV}$ Timing control signal is not delayed
		1	$\overline{ADV}$ Timing control signal is delayed of half GPMC_FCLK clock cycle

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 6-72. GPMC_CONFIG3_i Field Descriptions (continued)

Bit	Field	Value	Description
6-4	ADVAADMUXONTIME		ADV# assertion for first address phase when using the AAD-Multiplexed protocol
		0	0 GPMC_FCLK cycle
		1h	1 GPMC_FCLK cycle
		⋮	⋮
		7h	7 GPMC_FCLK cycles
3-0	ADVONTIME		ADV# assertion time from start cycle time
		0	0 GPMC_FCLK cycle
		1h	1 GPMC_FCLK cycle
		⋮	⋮
		Fh	15 GPMC_FCLK cycles

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.5.14 GPMC_CONFIG4_i

WE# and OE# signals timing parameter configuration.

Figure 6-65. GPMC_CONFIG4_i

31–29	28–24
Reserved	WEOFFTIME
R-0	R/W-0

23	22–20	19–16
WEEXTRADELAY	Reserved	WEONTIME
R/W-0	R-0	R/W-0

15–13	12–8
OEAADMUXOFFTIME	OEOFFTIME
R/W-0	R/W-0

7	6–4	3–0
OEEXTRA DELAY	OEAADMUXONTIME	OEONTIME
R/W-0	R/W-0	R/W-0

LEGEND: R = Read only; W1C = Write 1 to clear bit; -*n* = value after reset

Table 6-73. GPMC_CONFIG4_i Field Descriptions

Bit	Field	Value	Description
31-29	Reserved	0	Reserved
28-24	WEOFFTIME		WE# de-assertion time from start cycle time
		0	0 GPMC_FCLK cycle
		1h	1 GPMC_FCLK cycle
		⋮	⋮
		1Fh	31 GPMC_FCLK cycles
23	WEEXTRADELAY		WE# Add Extra Half GPMC.FCLK cycle
		0	$\overline{WE}$ Timing control signal is not delayed
		1	$\overline{WE}$ Timing control signal is delayed of half GPMC_FCLK clock cycle
22-20	Reserved	0	Reserved
19-16	WEONTIME		WE# assertion time from start cycle time
		0	0 GPMC_FCLK cycle
		1h	1 GPMC_FCLK cycle
		⋮	⋮
		Fh	15 GPMC_FCLK cycles
15-13	OEAADMUXOFFTIME		OE# de-assertion time for the first address phase in an AAD-Multiplexed access
		0	0 GPMC_FCLK cycle
		1h	1 GPMC_FCLK cycle
		⋮	⋮
		7h	7 GPMC_FCLK cycles
12-8	OEOFFTIME		OE# de-assertion time from start cycle time
		0	0 GPMC_FCLK cycle
		1h	1 GPMC_FCLK cycle
		⋮	⋮
		1Fh	31 GPMC_FCLK cycles
7	OEEXTRADELAY		OE# Add Extra Half GPMC.FCLK cycle
		0	$\overline{OE}$ Timing control signal is not delayed
		1	$\overline{OE}$ Timing control signal is delayed of half GPMC_FCLK clock cycle

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 6-73. GPMC_CONFIG4_i Field Descriptions (continued)

Bit	Field	Value	Description
6-4	OEAADMUXONTIME		OE# assertion time for the first address phase in an AAD-Multiplexed access
		0	0 GPMC_FCLK cycle
		1h	1 GPMC_FCLK cycle
		⋮	⋮
		7h	7 GPMC_FCLK cycles
3-0	OEONTIME		OE# assertion time from start cycle time
		0	0 GPMC_FCLK cycle
		1h	1 GPMC_FCLK cycle
		⋮	⋮
		Fh	15 GPMC_FCLK cycles

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.5.15 GPMC_CONFIG5_i

RdAccessTime and CycleTime timing parameters configuration.

Figure 6-66. GPMC_CONFIG5_i

31–28	27–24	23–21	20–16
Reserved	PAGEBURSTACCESSTIME	Reserved	RDACCESSTIME
R-0	R/W-0	R-0	R/W-0

15–13	12–8	7–5	4–0
Reserved	WRCYCLETIME	Reserved	RDCYCLETIME
R-0	R/W-0	R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 6-74. GPMC_CONFIG5_i Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Reserved
27-24	PAGEBURSTACCESSTIME		Delay between successive words in a multiple access
		0	0 GPMC_FCLK cycle
		1h	1 GPMC_FCLK cycle
		⋮	⋮
		Fh	15 GPMC_FCLK cycles
23-21	Reserved	0	Reserved
20-16	RDACCESSTIME		Delay between start cycle time and first data valid
		0	0 GPMC_FCLK cycle
		1h	1 GPMC_FCLK cycle
		⋮	⋮
		1Fh	31 GPMC_FCLK cycles
15-13	Reserved	0	Reserved
12-8	WRCYCLETIME		Total write cycle time
		0	0 GPMC_FCLK cycle
		1h	1 GPMC_FCLK cycle
		⋮	⋮
		1Fh	31 GPMC_FCLK cycles
7-5	Reserved	0	Reserved
4-0	RDCYCLETIME		Total read cycle time
		0	0 GPMC_FCLK cycle
		1h	1 GPMC_FCLK cycle
		⋮	⋮
		1Fh	31 GPMC_FCLK cycles

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.5.16 GPMC_CONFIG6_i

WrAccessTime, WrDataOnADmuxBus, Cycle2Cycle, and BusTurnAround parameters configuration

Figure 6-67. GPMC_CONFIG6_i

31–29	28–24	23–20	19–16
Reserved	WRACCESSTIME	Reserved	WRDATAONADMUXBUS
R-0	R/W-0	R-0	R/W-0

15–12	11–8
Reserved	CYCLE2CYCLEDELAY
R-0	R/W-0

7	6	5–4	3–0
CYCLE2CYCLE SAMECSEN	CYCLE2CYCLE DIFFCSEN	Reserved	BUSTURNAROUND
R/W-0	R/W-0	R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 6-75. GPMC_CONFIG6_i Field Descriptions

Bit	Field	Value	Description
31-29	Reserved	0	Reserved
28-24	WRACCESSTIME		Delay from StartAccessTime to the GPMC.FCLK rising edge corresponding the the GPMC.CLK rising edge used by the attached memory for the first data capture
		0	0 GPMC_FCLK cycle
		1h	1 GPMC_FCLK cycle
		⋮	⋮
		1Fh	31 GPMC_FCLK cycles
23-20	Reserved	0	Reserved
19-16	WRDATAONADMUXBUS	0-Fh	Specifies on which GPMC.FCLK rising edge the first data of the synchronous burst write is driven in the add/data multiplexed bus
15-12	Reserved	0	Reserved
11-8	CYCLE2CYCLEDELAY		Chip select high pulse delay between two successive accesses
		0	0 GPMC_FCLK cycle
		1h	1 GPMC_FCLK cycle
		⋮	⋮
		Fh	15 GPMC_FCLK cycles
7	CYCLE2CYCLESAMECSEN		Add Cycle2CycleDelay between two successive accesses to the same chip-select (any access type)
		0	No delay between the two accesses
		1	Add CYCLE2CYCLEDELAY
6	CYCLE2CYCLEDIFFCSEN		Add Cycle2CycleDelay between two successive accesses to a different chip-select (any access type)
		0	No delay between the two accesses
		1	Add CYCLE2CYCLEDELAY
5-4	Reserved	0	Reserved
3-0	BUSTURNAROUND		Bus turn around latency between two successive accesses to the same chip-select (read to write) or to a different chip-select (read to read and read to write)
		0	0 GPMC_FCLK cycle
		1h	1 GPMC_FCLK cycle
		⋮	⋮
		Fh	15 GPMC_FCLK cycles

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.5.17 GPMC_CONFIG7_i

Chip-select address mapping configuration.

Figure 6-68. GPMC_CONFIG7_i

31–16	15–12	11–8	7	6	5–0
Reserved	Reserved	MASKADDRESS	Rsvd	CSVALID	BASEADDRESS
R-0	R-0	R/W-0	R-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 6-76. GPMC_CONFIG7_i Field Descriptions

Bit	Field	Value	Description
31-12	Reserved	0	Reserved
11-8	MASKADDRESS		Chip-select mask address. Values not listed must be avoided as they create holes in the chip-select address space.
		0	Chip-select size of 256 Mbytes
		8h	Chip-select size of 128 Mbytes
		Ch	Chip-select size of 64 Mbytes
		Eh	Chip-select size of 32 Mbytes
		Fh	Chip-select size of 16 Mbytes
7	Reserved	0	Reserved
6	CSVALID		Chip-select enable (reset value is 1 for $\overline{CS}$[0] and 0 for $\overline{CS}$[1-7]).
		0	$\overline{CS}$ disabled
		1	$\overline{CS}$ enabled
5-0	BASEADDRESS	0-3Fh	Chip-select base address.
			CSi base address where i = 0 to 3 (16 Mbytes minimum granularity). Bits [5-0] corresponds to A29, A28, A27, A26, A25, and A24.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.5.18 GPMC_NAND_COMMAND_i

This register is not a true register, just an address location.

Figure 6-69. GPMC_NAND_COMMAND_i

31	0
GPMC_NAND_COMMAND_i	
W-0	

LEGEND: W = Write only; -n = value after reset

Table 6-77. GPMC_NAND_COMMAND_i Field Descriptions

Bit	Field	Value	Description
31-0	GPMC_NAND_COMMAND_i	0-FFFF FFFFh	Writing data at the GPMC_NAND_COMMAND_i location places the data as the NAND command value on the bus, using a regular asynchronous write access.

6.1.5.19 GPMC_NAND_ADDRESS_i

This register is not a true register, just an address location.

Figure 6-70. GPMC_NAND_ADDRESS_i

31	0
GPMC_NAND_ADDRESS_i	
W-0	

LEGEND: W = Write only; -n = value after reset

Table 6-78. GPMC_NAND_ADDRESS_i Field Descriptions

Bit	Field	Value	Description
31-0	GPMC_NAND_ADDRESS_i	0-FFFF FFFFh	Writing data at the GPMC_NAND_ADDRESS_i location places the data as the NAND partial address value on the bus, using a regular asynchronous write access.

6.1.5.20 GPMC_NAND_DATA_i

This register is not a true register, just an address location.

Figure 6-71. GPMC_NAND_DATA_i

31	0
GPMC_NAND_DATA_i	
R/W	

LEGEND: R/W = Read/Write; R = Read only; W = Write only; -n = value after reset

Table 6-79. GPMC_NAND_DATA_i Field Descriptions

Bit	Field	Value	Description
31-0	GPMC_NAND_DATA_i	0-FFFF FFFFh	Reading data from the GPMC_NAND_DATA_i location or from any location in the associated chip-select memory region activates an asynchronous read access.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.5.21 GPMC_PREFETCH_CONFIG1

Figure 6-72. GPMC_PREFETCH_CONFIG1

31	30–28	27	26–24
Reserved	CYCLEOPTIMIZATION	ENABLEOPTIMIZED ACCESS	ENGINECSSELECTOR
R-0	R/W-0	R/W-0	R/W-0

23	22–20	19–16
PFPWENROUNDROBIN	Reserved	PFPWWEIGHTEDPRIO
R/W-0	R-0	R/W-0

15	14–8
Reserved	FIFOTHRESHOLD
R-0	R/W-0

7	6	5–4	3	2	1	0
ENABLEENGINE	Reserved	WAITPINSELECTOR	SYNCHROMODE	DMAMODE	Reserved	ACCESSMODE
R/W-0	R-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0

LEGEND: R = Read only; -*n* = value after reset

Table 6-80. GPMC_PREFETCH_CONFIG1 Field Descriptions

Bit	Field	Value	Description
31	Reserved	0	Reserved
30-28	CYCLEOPTIMIZATION		Define the number of GPMC.FCLK cycles to be substracted from RdCycleTime, WrCycleTime, AccessTime, CSRdOffTime, CSWrOffTime, ADVRdOffTime, ADVWrOffTime, OEOffTime, WEOffTime
		0	0 GPMC_FCLK cycle
		1h	1 GPMC_FCLK cycle
		⋮	⋮
		7h	7 GPMC_FCLK cycles
27	ENABLEOPTIMIZEDACCESS		Enables access cycle optimization
		0	Access cycle optimization is disabled
		1	Access cycle optimization is enabled
26-24	ENGINECSSELECTOR		Selects the $\overline{CS}$ where Prefetch Postwrite engine is active
		0	$\overline{CS0}$
		1h	$\overline{CS1}$
		2h	$\overline{CS2}$
		3h	$\overline{CS3}$
		4h	$\overline{CS4}$
		5h	$\overline{CS5}$
		6h	$\overline{CS6}$
		7h	$\overline{CS7}$
23	PFPWENROUNDROBIN		Enables the PFPW RoundRobin arbitration
		0	Prefetch Postwrite engine round robin arbitration is disabled
		1	Prefetch Postwrite engine round robin arbitration is enabled
22-20	Reserved	0	Reserved
19-16	PFPWWEIGHTEDPRIO		When an arbitration occurs between a direct memory access and a PFPW engine access, the direct memory access is always serviced. If the PFPWEnRoundRobin is enabled,
		0	The next access is granted to the PFPW engine
		1h	The two next accesses are granted to the PFPW engine
		⋮	⋮
		Fh	The 16 next accesses are granted to the PFPW engine
15	Reserved	0	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 6-80. GPMC_PREFETCH_CONFIG1 Field Descriptions (continued)

Bit	Field	Value	Description
14-8	FIFOTHRESHOLD		Selects the maximum number of bytes read from the FIFO or written to the FIFO by the host on a DMA or interrupt request
		0	0 byte
		1h	1 byte
		⋮	⋮
		40h	64 bytes
7	ENABLEENGINE		Enables the Prefetch Postwite engine
		0	Prefetch Postwrite engine is disabled
		1	Prefetch Postwrite engine is enabled
6	Reserved	0	Reserved
5-4	WAITPINSELECTOR		Select which WAIT pin edge detector should start the engine in synchronized mode
		0	Selects Wait0EdgeDetection
		1h	Selects Wait1EdgeDetection
		2h	Reserved
		3h	Reserved
3	SYNCHROMODE		Selects when the engine starts the access to CS
		0	Engine starts the access to CS as soon as STARTENGINE is set
		1	Engine starts the access to CS as soon as STARTENGINE is set AND wait to non wait edge detection on the selected wait pin
2	DMAMODE		Selects interrupt synchronization or DMA request synchronization
		0	Interrupt synchronization is enabled. Only interrupt line will be activated on FIFO threshold crossing.
		1	DMA request synchronization is enabled. A DMA request protocol is used.
1	Reserved	0	Reserved
0	ACCESSMODE		Selects pre-fetch read or write posting accesses
		0	Prefetch read mode
		1	Write-posting mode

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.5.22 GPMC_PREFETCH_CONFIG2

Figure 6-73. GPMC_PREFETCH_CONFIG2

31–16	15–14	13–0
Reserved	Reserved	TRANSFERCOUNT
R-0	R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 6-81. GPMC_PREFETCH_CONFIG2 Field Descriptions

Bit	Field	Value	Description
31-14	Reserved	0	Reserved
13-0	TRANSFERCOUNT		Selects the number of bytes to be read or written by the engine to the selected $\overline{CS}$
		0	0 byte
		1h	1 byte
		⋮	⋮
		2000h	8 Kbytes

6.1.5.23 GPMC_PREFETCH_CONTROL

Figure 6-74. GPMC_PREFETCH_CONTROL

31–16	15–1	0
Reserved	Reserved	STARTENGINE
R-0	R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 6-82. GPMC_PREFETCH_CONTROL Field Descriptions

Bit	Field	Value	Description
31-1	Reserved	0	Reserved
0	STARTENGINE		Resets the FIFO pointer and starts the engine
		R0	Engine is stopped
		W0	Stops the engine
		R1	Engine is running
		W1	Resets the FIFO pointer to 0 in prefetch mode and 40h in postwrite mode and starts the engine

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.5.24 GPMC_PREFETCH_STATUS

Figure 6-75. GPMC_PREFETCH_STATUS

31	30–24	23–17	16
Rsvd	FIFOPOINTER	Reserved	FIFOTHRESHOLDSTATUS
R-0	R-0	R-0	R-0

15–14	13–0
Reserved	COUNTVALUE
R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 6-83. GPMC_PREFETCH_STATUS Field Descriptions

Bit	Field	Value	Description
31	Reserved	0	Reserved
30-24	FIFOPOINTER	0 ⋮ 40h	0 byte available to be read or 0 free empty place to be written ⋮ 64 bytes available to be read or 64 empty places to be written
23-17	Reserved	0	Reserved
16	FIFOTHRESHOLDSTATUS		Set when FIFOPointer exceeds FIFOThreshold value
		0	FIFOPointer smaller or equal to FIFOThreshold. Writing to this bit has no effect
		1	FIFOPointer greater than FIFOThreshold. Writing to this bit has no effect
15-14	Reserved	0	Reserved
13-0	COUNTVALUE		Number of remaining bytes to be read or to be written by the engine according to the TransferCount value
		0	0 byte remaining to be read or to be written
		1h	1 byte remaining to be read or to be writte
		⋮	⋮
		2000h	8 Kbytes remaining to be read or to be written

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.5.25 GPMC_ECC_CONFIG

Figure 6-76. GPMC_ECC_CONFIG

31	17	16
Reserved		ECCALGORITHM
R-0		R/W-0

15	14	13	12	11	8
Reserved		ECCBCHTSEL		ECCWRAPMODE	
R-0		R/W-0		R/W-0	

7	6	4	3	1	0
ECC16B	ECCTOPSECTOR		ECCCS		ECCENABLE
R/W-0	R/W-0		R/W-0		R/W-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 6-84. GPMC_ECC_CONFIG Field Descriptions

Bit	Field	Value	Description
31-17	Reserved	0	Reserved
16	ECCALGORITHM		ECC algorithm used
		0	Hamming code
		1	BCH code
15-14	Reserved	0	Reserved
13-12	ECCBCHTSEL		Error correction capability used for BCH
		0	Up to 4 bits error correction (t = 4)
		1h	Up to 8 bits error correction (t = 8)
		2h	Up to 16 bits error correction (t = 16)
		3h	Reserved
11-8	ECCWRAPMODE	0-Fh	Spare area organization definition for the BCH algorithm. See the BCH syndrome/parity calculator module functional specification for more details
7	ECC16B		Selects an ECC calculated on 16 columns
		0	ECC calculated on 8 columns
		1	ECC calculated on 16 columns
6-4	ECCTOPSECTOR		Number of sectors to process with the BCH algorithm
		0	1 sector (512kB page)
		1h	2 sectors
		3h	4 sectors (2kB page)
		⋮	⋮
		7h	8 sectors (4kB page)
3-1	ECCCS		Selects the Chip-select where ECC is computed
		0	Chip-select 0
		1h	Chip-select 1
		2h	Chip-select 2
		3h	Chip-select 3
		4h	Chip-select 4
		5h	Chip-select 5
		6h-7h	Reserved
0	ECCENABLE		Enables the ECC feature
		0	ECC disabled
		1	ECC enabled

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.5.26 GPMC_ECC_CONTROL

Figure 6-77. GPMC_ECC_CONTROL

31–16	15–9	8	7–4	3–0
Reserved	Reserved	ECCCLEAR	Reserved	ECCPOINTER
R-0	R-0	R/W-0	R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 6-85. GPMC_ECC_CONTROL Field Descriptions

Bit	Field	Value	Description
31-9	Reserved	0	Reserved
8	ECCCLEAR		Clear all ECC result registers
		R0	All reads return 0
		W0	Ignored
		W1	Clears all ECC result registers
7-4	Reserved	0	Reserved
3-0	ECCPOINTER	0-Fh	Selects ECC result register (Reads to this field give the dynamic position of the ECC pointer - Writes to this field select the ECC result register where the first ECC computation will be stored). Writing values not listed disables the ECC engine (ECCEnable bit of GPMC_ECC_CONFIG cleared to 0).
		0	Writing 0 disables the ECC engine (ECCENABLE bit of GPMC_ECC_CONFIG cleared to 0)
		1h	ECC result register 1 selected
		2h	ECC result register 2 selected
		3h	ECC result register 3 selected
		4h	ECC result register 4 selected
		5h	ECC result register 5 selected
		6h	ECC result register 6 selected
		7h	ECC result register 7 selected
		8h	ECC result register 8 selected
		9h	ECC result register 9 selected

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.5.27 GPMC_ECC_SIZE_CONFIG

Figure 6-78. GPMC_ECC_SIZE_CONFIG

31 30	29 22	21 20	19 16
Reserved	ECCSIZE1	Reserved	ECCSIZE0
R-0	R/W-0	R-0	R/W-0

15 12	11 9	8
ECCSIZE0	Reserved	ECC9RESULTSIZE
R/W-0	R-0	R/W-0

7	6	5	4	3	2	1	0
ECC8RESULTSIZE	ECC7RESULTSIZE	ECC6RESULTSIZE	ECC5RESULTSIZE	ECC4RESULTSIZE	ECC3RESULTSIZE	ECC2RESULTSIZE	ECC1RESULTSIZE
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 6-86. GPMC_ECC_SIZE_CONFIG Field Descriptions

Bit	Field	Value	Description
31-30	Reserved	0	Reserved
29-22	ECCSIZE1		Defines ECC size 1
		0	2 Bytes
		1h	4 Bytes
		2h	6 Bytes
		3h	8 Bytes
		⋮	⋮
		FFh	512 Bytes
21-20	Reserved	0	Reserved
19-12	ECCSIZE0		Defines ECC size 0
		0	2 Bytes
		1h	4 Bytes
		2h	6 Bytes
		3h	8 Bytes
		⋮	⋮
		FFh	512 Bytes
11-9	Reserved	0	Reserved
8	ECC9RESULTSIZE		Selects ECC size for ECC 9 result register
		0	ECCSIZE0 selected
		1	ECCSIZE1 selected
7	ECC8RESULTSIZE		Selects ECC size for ECC 8 result register
		0	ECCSIZE0 selected
		1	ECCSIZE1 selected
6	ECC7RESULTSIZE		Selects ECC size for ECC 7 result register
		0	ECCSIZE0 selected
		1	ECCSIZE1 selected
5	ECC6RESULTSIZE		Selects ECC size for ECC 6 result register
		0	ECCSIZE0 selected
		1	ECCSIZE1 selected
4	ECC5RESULTSIZE		Selects ECC size for ECC 5 result register
		0	ECCSIZE0 selected
		1	ECCSIZE1 selected

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 6-86. GPMC_ECC_SIZE_CONFIG Field Descriptions (continued)

Bit	Field	Value	Description
3	ECC4RESULTSIZE		Selects ECC size for ECC 4 result register
		0	ECCSIZE0 selected
		1	ECCSIZE1 selected
2	ECC3RESULTSIZE		Selects ECC size for ECC 3 result register
		0	ECCSIZE0 selected
		1	ECCSIZE1 selected
1	ECC2RESULTSIZE		Selects ECC size for ECC 2 result register
		0	ECCSIZE0 selected
		1	ECCSIZE1 selected
0	ECC1RESULTSIZE		Selects ECC size for ECC 1 result register
		0	ECCSIZE0 selected
		1	ECCSIZE1 selected

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.5.28 GPMC_ECCj_RESULT

Figure 6-79. GPMC_ECCj_RESULT

31–28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	P2048O	P1024O	P512O	P256O	P128O	P64O	P32O	P16O	P8O	P4O	P2O	P1O
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

15–12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	P2048E	P1024E	P512E	P256E	P128E	P64E	P32E	P16E	P8E	P4E	P2E	P1E
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 6-87. GPMC_ECCj_RESULT Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Reserved
27	P2048O	0-1	Odd Row Parity bit 2048, only used for ECC computed on 512 Bytes
26	P1024O	0-1	Odd Row Parity bit 1024
25	P512O	0-1	Odd Row Parity bit 512
24	P256O	0-1	Odd Row Parity bit 256
23	P128O	0-1	Odd Row Parity bit 128
22	P64O	0-1	Odd Row Parity bit 64
21	P32O	0-1	Odd Row Parity bit 32
20	P16O	0-1	Odd Row Parity bit 16
19	P8O	0-1	Odd Row Parity bit 8
18	P4O	0-1	Odd Column Parity bit 4
17	P2O	0-1	Odd Column Parity bit 2
16	P1O	0-1	Odd Column Parity bit 1
15-12	Reserved	0	Reserved
11	P2048E	0-1	Even Row Parity bit 2048, only used for ECC computed on 512 Bytes
10	P1024E	0-1	Even Row Parity bit 1024
9	P512E	0-1	Even Row Parity bit 512
8	P256E	0-1	Even Row Parity bit 256
7	P128E	0-1	Even Row Parity bit 128
6	P64E	0-1	Even Row Parity bit 64
5	P32E	0-1	Even Row Parity bit 32
4	P16E	0-1	Even Row Parity bit 16
3	P8E	0-1	Even Row Parity bit 8
2	P4E	0-1	Even Column Parity bit 4
1	P2E	0-1	Even Column Parity bit 2
0	P1E	0-1	Even Column Parity bit 1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.5.29 GPMC_BCH_RESULT0_i

Figure 6-80. GPMC_BCH_RESULT0_i

31	0
BCH_RESULT0_i	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 6-88. GPMC_BCH_RESULT0_i Field Descriptions

Bit	Field	Value	Description
31-0	BCH_RESULT0_i	0-FFFF FFFFh	BCH ECC result, bits 0 to 31

6.1.5.30 GPMC_BCH_RESULT1_i

Figure 6-81. GPMC_BCH_RESULT1_i

31	0
BCH_RESULT1_i	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 6-89. GPMC_BCH_RESULT1_i Field Descriptions

Bit	Field	Value	Description
31-0	BCH_RESULT1_i	0-FFFF FFFFh	BCH ECC result, bits 32 to 63

6.1.5.31 GPMC_BCH_RESULT2_i

Figure 6-82. GPMC_BCH_RESULT2_i

31	0
BCH_RESULT2_i	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 6-90. GPMC_BCH_RESULT2_i Field Descriptions

Bit	Field	Value	Description
31-0	BCH_RESULT2_i	0-FFFF FFFFh	BCH ECC result, bits 64 to 95

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.5.32 GPMC_BCH_RESULT3_i

Figure 6-83. GPMC_BCH_RESULT3_i

31	0
BCH_RESULT3_i	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 6-91. GPMC_BCH_RESULT3_i Field Descriptions

Bit	Field	Value	Description
31-0	BCH_RESULT3_i	0-FFFF FFFFh	BCH ECC result, bits 96 to 127

6.1.5.33 GPMC_BCH_SWDATA

This register is used to directly pass data to the BCH ECC calculator without accessing the actual NAND flash interface.

Figure 6-84. GPMC_BCH_SWDATA

31	16
Reserved	
R-0	

15	0
BCH_DATA	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 6-92. GPMC_BCH_SWDATA Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15-0	BCH_DATA	0-FFFFh	Data to be included in the BCH calculation. Only bits 0 to 7 are taken into account, if the calculator is configured to use 8 bits data (GPMC_ECC_CONFIG[7] ECC16B = 0).

6.1.5.34 GPMC_BCH_RESULT4_i

Figure 6-85. GPMC_BCH_RESULT4_i

31	0
BCH_RESULT4_i	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 6-93. GPMC_BCH_RESULT4_i Field Descriptions

Bit	Field	Value	Description
31-0	BCH_RESULT4_i	0-FFFF FFFFh	BCH ECC result, bits 128 to 159

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.1.5.35 GPMC_BCH_RESULT5_i

Figure 6-86. GPMC_BCH_RESULT5_i

31	0
BCH_RESULT5_i	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 6-94. GPMC_BCH_RESULT5_i Field Descriptions

Bit	Field	Value	Description
31-0	BCH_RESULT5_i	0-FFFF FFFFh	BCH ECC result, bits 160 to 191

6.1.5.36 GPMC_BCH_RESULT6_i

Figure 6-87. GPMC_BCH_RESULT6_i

31	0
BCH_RESULT6_i	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 6-95. GPMC_BCH_RESULT6_i Field Descriptions

Bit	Field	Value	Description
31-0	BCH_RESULT6_i	0-FFFF FFFFh	BCH ECC result, bits 192 to 207

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.2 OCMC-RAM

6.2.1 Introduction

6.2.1.1 OCMC RAM Features

The on-chip memory controller consists of two separate modules that are OCP to ACE wrappers for ACE memories. The first wrapper is for a ROM; the second is for a RAM. Each wrapper has its own dedicated interface to the L3 interconnect. This device uses only the RAM wrapper which supports both BRGS single port and GLG single port CRAM.

- 32- or 64-bit width
- Initial latency max 2 cycles (due to OCP to memory core wrapper).
- Multiple memory bank control based on address MSBs
- Full OCP IP 2.0 Burst support. No wait state.

6.2.1.2 Unsupported OCMC RAM Features

For this device, the OCMC RAM implementation does not support parity.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.2.2 Integration

This device includes a single instantiation of the on-chip memory controller interfacing to a single 64-K bank of RAM.

Figure 6-88. OCMC RAM Integration

6.2.2.1 OCMC RAM Connectivity Attributes

The general connectivity attributes for the OCMC RAM modules are summarized in Table 6-96.

Table 6-96. OCMC RAM Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_L3_GCLK
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	None
DMA Requests	None
Physical Address	L3 Fast slave port

6.2.2.2 OCMC RAM Clock and Reset Management

The OCMC module uses a single clock for the module and its OCP interface.

Table 6-97. OCMC RAM Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
prcm_ocmc_clock Interface / Functional clock	200 MHz	CORE_CLKOUTM4	pd_per_l3_gclk From PRCM

6.2.2.3 OCMC RAM Pin List

The OCMC RAM module does not include any external interface pins.

6.2.3 Functional Description

6.2.4 Use Cases

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.2.5 Registers

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3 EMIF

6.3.1 Introduction

6.3.1.1 Features

The general features of the EMIF module are as follows:

- 16-bit and 32-bit data path to external SDRAM memory
  - Unused DID Data Macros disabled in 16-bit mode
- One 128-bit OCPIP 2.2 interface
- Support for the following memory types:
  - MDDR (LPDDR1)
  - DDR2
  - DDR3 (stretch goal)

External memory configurations supported:

- Memory device capacity
  - 64Mb, 128Mb, 256Mb, 512Mb, 1Gb and 2Gb devices support
- Flexible bank/row/column/chip-select address multiplexing schemes
- Support for 2 independent chip select, with their corresponding register sets, and independent page tracking
- Device driver strength feature for mobile DDR supported
- Supports following CAS latencies:
  - DDR2 => 2, 3, 4, 5, 6, and 7
  - DDR3 => 5, 6, 7, 8, 9, 10, and 11
  - mDDR => 2, 3, and 4
- Supports following number of internal banks:
  - DDR2 => 1, 2, 4, and 8
  - DDR3 => 1, 2, 4, and 8
  - mDDR => 1, 2, and 4
- Supports 256, 512, 1024, and 2048-word page sizes
- Supports burst length of 8 (sequential burst)
- Write/read leveling/calibration and data eye training in conjunction with GS70 DID
- Self Refresh and Power-Down modes for low power:
  - Flexible OCP to DDR address mapping to support Partial Array Self Refresh in LPDDR1, DDR2 and DDR3.
  - Temperature Controlled Self Refresh for LPDDR1 and DDR3 having on-chip temperature sensor.
- Periodic ZQ calibration for DDR3
- ODT on DDR2 and DDR3
- Prioritized refresh scheduling
- Programmable SDRAM refresh rate and backlog counter
- Programmable SDRAM timing parameters
- SDRAM Data bus obfuscation for security without additional cycle latency
- Big and little endian modes

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.1.2 Unsupported EMIF Features

The following EMIF4DC module features are not supported in this device.

Table 6-98. Unsupported EMIF Features

Feature	Reason
32-bit data	Only 16 bits pinned out
Multiple DDR banks	Only 1 CS/ODT pinned out
DDR3	DDR3 is fully supported by the EMIF/Phy and pinout but achieving the required DDR3 minimum frequency is a stretch goal for the device.
DDR2 CAS Latency 2	Not supported by DID

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.2 Integration

6.3.2.1 EMIF Connectivity Attributes

The general connectivity attributes for the EMIF are shown in Table 6-99.

Table 6-99. EMIF Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_L3)GCLK (OCP) PD_PER_EMIF_GCLK (Func)
Reset Signals	USB_POR_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	1 interrupt to Sabertooth
DMA Requests	None
Physical Address	L3 Fast Slave Port

6.3.2.2 EMIF Clock Management

The EMIF4 OCP interface (ocp_clk) is clocked by the L3 Fast clock sourced from the Core PLL. The DDR Command and Data macros are clocked by the DDR PLL. The PRCM divides this clock by two to create the EMIF functional clock (m_clk).

The OCP and functional clocks may be asynchronous because synchronization is managed in the EMIF4 internal FIFO (EMIF4 is set in asynchronous mode).

Table 6-100. EMIF Clock Signals

Clock Signal	Max Freq	Reference Source	Comments
ocp_clk (Interface clock)	200 Mhz	CORE_CLKOUTM4	pd_per_l3_gclk From PRCM
m_clk (EMIF functional clock)	100 Mhz	DDR PLL CLKOUT / 2	pd_per_emif_gclk From PRCM
cmd0_dfi_clk cmd1_dfi_clk cmd2_dfi_clk data0_dfi_clk data1_dfi_clk (Macro clocks)	200 MHz	DDR PLL CLKOUT	clkout_po From DDR PLL

6.3.2.3 EMIF Pin List

The EMIF/DDR external interface signals are shown in Table 6-101.

Table 6-101. EMIF Pin List

Pin	Type	Description
DDR_CK DDR_NCK	O	Differential clock pair
DDR_CKE	O	Clock enable
DDR_CSn0	O	Chip select
DDR_RASn	O	Row address strobe
DDR_CASn	O	Column address strobe
DDR_WEn	O	Write enable
DDR_BA[2:0]	O	Bank address

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 6-101. EMIF Pin List (continued)

Pin	Type	Description
DDR_A[15:0]	O	Row/column address
DDR_DQS[1:0]	I/O	Data strobes
DDR_DQSn[1:0]	I/O	Complimentary data strobes
DDR_DQM[1:0]	O	Data masks
DDR_D[15:0]	I/O	Data
DDR_ODT	O	On-die termination
DDR_RESET	O	DDR device reset
DDR_VREF	I	I/O Voltage reference

6.3.3 Functional Description

6.3.4 Use Cases

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5 EMIF4D Registers

Table 6-102 lists the memory-mapped registers for the EMIF4D. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 6-102 should be considered as reserved locations and the register contents should not be modified.

Table 6-102. EMIF4D REGISTERS

Offset	Name	Register Description	Section
0h	EMIF_MOD_ID_REV		Section 6.3.5.1
64h	IODFT_CTRL_MISR_RSLT		Section 6.3.5.2
68h	IODFT_ADDR_MISR_RSLT		Section 6.3.5.3
6Ch	IODFT_DATA_MISR_RSLT_1		Section 6.3.5.4
70h	IODFT_DATA_MISR_RSLT_2		Section 6.3.5.5
74h	IODFT_DATA_MISR_RSLT_3		Section 6.3.5.6
Ch	SDRAM_CONFIG_2		Section 6.3.5.7
80h	PERF_CNT_1		Section 6.3.5.8
84h	PERF_CNT_2		Section 6.3.5.9
88h	PERF_CNT_CFG		Section 6.3.5.10
8Ch	PERF_CNT_SEL		Section 6.3.5.11
90h	PERF_CNT_TIM		Section 6.3.5.12
98h	READ_IDLE_CTRL		Section 6.3.5.13
9Ch	READ_IDLE_CTRL_SHDW		Section 6.3.5.14
10h	SDRAM_REF_CTRL		Section 6.3.5.15
A0h	IRQ_EOI		Section 11.5.54
A4h	IRQSTATUS_RAW_SYS		Section 6.3.5.17
A8h	IRQSTATUS_RAW_LL		Section 6.3.5.18
ACh	IRQSTATUS_SYS		Section 6.3.5.19
B0h	IRQSTATUS_LL		Section 6.3.5.20
B4h	IRQENABLE_SET_SYS		Section 6.3.5.21
B8h	IRQENABLE_SET_LL		Section 6.3.5.22
BCh	IRQENABLE_CLR_SYS		Section 6.3.5.23
C0h	IRQENABLE_CLR_LL		Section 6.3.5.24
14h	SDRAM_REF_CTRL_SHDW		Section 6.3.5.25
C8h	ZQ_CONFIG		Section 6.3.5.26
CCh	TEMP_ALERT_CONFIG		Section 6.3.5.27
D0h	OCP_ERR_LOG		Section 6.3.5.28
E4h	DDR_PHY_CTRL_1		Section 6.3.5.29
E8h	DDR_PHY_CTRL_1_SHDW		Section 6.3.5.30
ECh	DDR_PHY_CTRL_2		Section 6.3.5.31
18h	SDRAM_TIM_1		Section 6.3.5.32
1Ch	SDRAM_TIM_1_SHDW		Section 6.3.5.33
20h	SDRAM_TIM_2		Section 6.3.5.34
24h	SDRAM_TIM_2_SHDW		Section 6.3.5.35
4h	STATUS		Section 6.3.5.36
28h	SDRAM_TIM_3		Section 6.3.5.37
2Ch	SDRAM_TIM_3_SHDW		Section 6.3.5.38
30h	LPDDR2_NVM_TIM		Section 6.3.5.39
34h	LPDDR2_NVM_TIM_SHDW		Section 6.3.5.40
38h	PWR_MGMT_CTRL		Section 6.3.5.41
3Ch	PWR_MGMT_CTRL_SHDW		Section 6.3.5.42
40h	LPDDR2_MODE_REG_DATA		Section 6.3.5.43

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 6-102. EMIF4D REGISTERS (continued)

Offset	Name	Register Description	Section
8h	SDRAM_CONFIG		Section 6.3.5.44
50h	LPDDR2_MODE_REG_CFG		Section 6.3.5.45
54h	OCP_CONFIG		Section 6.3.5.46
58h	OCP_CFG_VAL_1		Section 6.3.5.47
5Ch	OCP_CFG_VAL_2		Section 6.3.5.48
60h	IODFT_TLGC		Section 6.3.5.49

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.1 EMIF_MOD_ID_REV Register (offset = 0h)

EMIF_MOD_ID_REV is shown in Figure 6-89 and described in Table 6-103.

Figure 6-89. EMIF_MOD_ID_REV Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-103. EMIF_MOD_ID_REV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	reg_scheme	R	0h	Used to distinguish between old and current schemes.
29-28	Reserved	R	0h	
27-16	reg_module_id	R	4h	EMIF module ID.
15-11	reg_rtl_version	R	1h	RTL Version.
10-8	reg_major_revision	R	0h	Major Revision.
7-6	Reserved	R	0h	
5-0	reg_minor_revision	R	2h	Minor Revision.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.2 IODFT_CTRL_MISR_RSLT Register (offset = 64h)

IODFT_CTRL_MISR_RSLT is shown in Figure 6-90 and described in Table 6-104.

Figure 6-90. IODFT_CTRL_MISR_RSLT Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-104. IODFT_CTRL_MISR_RSLT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	R	0h	
25-16	reg_dqm_tlmr	R	0h	This contains the MISR result signature of a given test after the download function is executed. This result is for the DQM signals.
15-11	Reserved	R	0h	
10-0	reg_ctl_tlmr	R	0h	This contains the MISR result signature of a given test after the download function is executed. This result is for the control signals.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.3 IODFT_ADDR_MISR_RSLT Register (offset = 68h)

IODFT_ADDR_MISR_RSLT is shown in Figure 6-91 and described in Table 6-105.

Figure 6-91. IODFT_ADDR_MISR_RSLT Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-105. IODFT_ADDR_MISR_RSLT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	Reserved	R	0h	
20-0	reg_addr_tlmr	R	0h	This contains the MISR result signature of a given test after the download function is executed. This result is for the address signals.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.4 IODFT_DATA_MISR_RSLT_1 Register (offset = 6Ch)

IODFT_DATA_MISR_RSLT_1 is shown in Figure 6-92 and described in Table 6-106.

Figure 6-92. IODFT_DATA_MISR_RSLT_1 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-106. IODFT_DATA_MISR_RSLT_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	reg_data_tlmr_31_0	R	0h	This contains the least significant bits of the MISR result signature of a given test after the download function is executed. This result is for data bus.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.5 IODFT_DATA_MISR_RSLT_2 Register (offset = 70h)

IODFT_DATA_MISR_RSLT_2 is shown in Figure 6-93 and described in Table 6-107.

Figure 6-93. IODFT_DATA_MISR_RSLT_2 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-107. IODFT_DATA_MISR_RSLT_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	reg_data_tlmr_63_32	R	0h	This contains the middle bits of the MISR result signature of a given test after the download function is executed. This result is for data bus.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.6 IODFT_DATA_MISR_RSLT_3 Register (offset = 74h)

IODFT_DATA_MISR_RSLT_3 is shown in Figure 6-94 and described in Table 6-108.

Figure 6-94. IODFT_DATA_MISR_RSLT_3 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-108. IODFT_DATA_MISR_RSLT_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	Reserved	R	0h	
2-0	reg_data_tlmr_66_64	R	0h	This contains the most significant bits of the MISR result signature of a given test after the download function is executed. This result is for data bus.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.7 SDRAM_CONFIG_2 Register (offset = Ch)

SDRAM_CONFIG_2 is shown in Figure 6-95 and described in Table 6-109.

Figure 6-95. SDRAM_CONFIG_2 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-109. SDRAM_CONFIG_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	Reserved	R	0h	
30	reg_cs1nvmen	R/W	0h	CS1 LPDDR2-NVM enable. Set to 1 if LPDDR2-NVM is connected to CS1. This bit will automatically be set to 0 if reg_sdram_type field in the SDRAM Config register is not set to LPDDR2.
29-28	Reserved	R	0h	
27	reg_ebank_pos	R/W	0h	External bank position. Set to 0 to assign external bank address bits from lower OCP address as shown in tables. Set to 1 to assign external bank address bits from higher OCP address bits as shown in tables.
26-6	Reserved	R	0h	
5-4	reg_rdbnum	R/W	0h	Row Buffer setup. Defines number of row buffers inside connected LPDDR2-NVM devices. Set to 0 for 1 row buffer, set to 1 for 2 row buffers, set to 2 for 4 row buffers, and set to 3 for 8 row buffers. All other values are reserved.
3	Reserved	R	0h	
2-0	reg_rdbsize	R/W	0h	Row Data Buffer Size. Defines the row data buffer size of connected LPDDR2-NVM devices. Set to 0 for 32 bytes, set to 1 for 64 bytes, set to 2 for 128 bytes, set to 3 for 256 bytes, set to 4 for 512 bytes, set to 5 for 1024 bytes, set to 6 for 2048 bytes, and set to 7 for 4096 bytes.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.8 PERF_CNT_1 Register (offset = 80h)

PERF_CNT_1 is shown in Figure 6-96 and described in Table 6-110.

Figure 6-96. PERF_CNT_1 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-110. PERF_CNT_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	reg_counter1	R	0h	32-bit counter that can be configured as specified in the Performance Counter Config Register and Performance Counter Master Region Select Register.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.9 PERF_CNT_2 Register (offset = 84h)

PERF_CNT_2 is shown in Figure 6-97 and described in Table 6-111.

Figure 6-97. PERF_CNT_2 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-111. PERF_CNT_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	reg_counter2	R	0h	32-bit counter that can be configured as specified in the Performance Counter Config Register and Performance Counter Master Region Select Register.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.10 PERF_CNT_CFG Register (offset = 88h)

PERF_CNT_CFG is shown in Figure 6-98 and described in Table 6-112.

Figure 6-98. PERF_CNT_CFG Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-112. PERF_CNT_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31	reg_cntr2_mconnid_en	R/W	0h	MConnID filter enable for Performance Counter 2 register.
30	reg_cntr2_region_en	R/W	0h	Chip Select filter enable for Performance Counter 2 register.
29-20	Reserved	R	0h	
19-16	reg_cntr2_cfg	R/W	0h	Filter configuration for Performance Counter 2. Refer to table for details.
15	reg_cntr1_mconnid_en	R/W	0h	MConnID filter enable for Performance Counter 1 register.
14	reg_cntr1_region_en	R/W	0h	Chip Select filter enable for Performance Counter 1 register.
13-4	Reserved	R	0h	
3-0	reg_cntr1_cfg	R/W	0h	Filter configuration for Performance Counter 1. Refer to table for details.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.11 PERF_CNT_SEL Register (offset = 8Ch)

PERF_CNT_SEL is shown in Figure 6-99 and described in Table 6-113.

Figure 6-99. PERF_CNT_SEL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-113. PERF_CNT_SEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	reg_mconnid2	R/W	0h	MConnID for Performance Counter 2 register.
23-18	Reserved	R	0h	
17-16	reg_region_sel2	R/W	0h	MAddrSpace for Performance Counter 2 register.
15-8	reg_mconnid1	R/W	0h	MConnID for Performance Counter 1 register.
7-2	Reserved	R	0h	
1-0	reg_region_sel1	R/W	0h	MAddrSpace for Performance Counter 1 register.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.12 PERF_CNT_TIM Register (offset = 90h)

PERF_CNT_TIM is shown in Figure 6-100 and described in Table 6-114.

Figure 6-100. PERF_CNT_TIM Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-114. PERF_CNT_TIM Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	reg_total_time	R	0h	32-bit counter that continuously counts number for m_clk cycles elapsed after EMIF is brought out of reset.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.13 READ_IDLE_CTRL Register (offset = 98h)

READ_IDLE_CTRL is shown in Figure 6-101 and described in Table 6-115.

Figure 6-101. READ_IDLE_CTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-115. READ_IDLE_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	Reserved	R	0h	
19-16	reg_read_idle_len	R/W	4h	The Read Idle Length field determines the minimum size (reg_read_idle_len-1 clock cycles) of Read Idle window for the read idle detection as well as the force read idle time.
15-9	Reserved	R	0h	
8-0	reg_read_idle_interval	R/W	0h	The Read Idle Interval field determines the maximum interval ((reg_read_idle_interval-1)*64 clock cycles) between read idle detections or force. A value of zero disables the read idle function.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.14 READ_IDLE_CTRL_SHDW Register (offset = 9Ch)

READ_IDLE_CTRL_SHDW is shown in Figure 6-102 and described in Table 6-116.

Figure 6-102. READ_IDLE_CTRL_SHDW Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-116. READ_IDLE_CTRL_SHDW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	Reserved	R	0h	
19-16	reg_read_idle_len_shdw	R/W	4h	Shadow field for reg_read_idle_len. This field is loaded into reg_read_idle_len field in Read Idle Control register when SIdleAck is asserted;
15-9	Reserved	R	0h	
8-0	reg_read_idle_interval_shdw	R/W	0h	Shadow field for reg_read_idle_interval. This field is loaded into reg_read_idle_interval field in Read Idle Control register when SIdleAck is asserted.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.15 SDRAM_REF_CTRL Register (offset = 10h)

SDRAM_REF_CTRL is shown in Figure 6-103 and described in Table 6-117.

Figure 6-103. SDRAM_REF_CTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 6-117. SDRAM_REF_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	reg_initref_dis	R/W	0h	Initialization and Refresh disable. When set to 1, EMIF will disable SDRAM initialization and refreshes, but will carry out SDRAM write/read transactions.
30	Reserved	R	0h	
29	reg_srt	R/W	0h	DDR3 Self Refresh temperature range. Set to 0 for normal operating temperature range and set to 1 for extended operating temperature range when the reg_asr field is set to 0. This bit must be set to 0 if the reg_asr field is set to 1. A write to this field will cause the EMIF to start the SDRAM initialization sequence.
28	reg_asr	R/W	0h	DDR3 Auto Self Refresh enable. Set to 1 for auto Self Refresh enable. Set to 0 for manual Self Refresh reference indicated by the reg_srt field. A write to this field will cause the EMIF to start the SDRAM initialization sequence.
27	Reserved	R	0h	
26-24	reg_pasr	R/W	0h	Partial Array Self Refresh. These bits get loaded into the Extended Mode Register of an LPDDR1 or DDR3 during initialization. For LPDDR1, set to 0 for full array, set to 1 for 1/2 array, set to 2 for 1/4 array, set to 5 for 1/8 array, and set to 6 for 1/16 array to be refreshed. For DDR3, set to 0 for full array, set to 1 or 5 for 1/2 array, set to 2 or 6 for 1/4 array, set to 3 or 7 for 1/8 array, and set to 4 for 3/4 array to be refreshed. All other values are reserved. A write to this field will cause the EMIF to start the SDRAM initialization sequence.
23-16	Reserved	R	0h	
15-0	reg_refresh_rate	R/W	0h	Refresh Rate. Value in this field is used to define the rate at which connected SDRAM devices will be refreshed. SDRAM refresh rate = EMIF rate / reg_refresh_rate where EMIF rate is equal to m_clk rate. If reg_refresh_rate andlt; (8*reg_t_rfc)+reg_t_rp+reg_t_rcd+20 then it will be loaded with (8*reg_t_rfc)+reg_t_rp+reg_t_rcd+20. This is done to avoid lock-up situations when illegal values are programmed.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.16 IRQ_EOI Register (offset = A0h)

IRQ_EOI is shown in Figure 11-58 and described in Table 11-59.

Figure 6-104. IRQ_EOI Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-118. IRQ_EOI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R	0h	
0	reg_eoi	R/W	0h	Software End Of Interrupt (EOI) control. Write 0 for system OCP interrupt and write 1 for low-latency OCP interrupt. This field always reads 0 (no EOI memory).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.17 IRQSTATUS_RAW_SYS Register (offset = A4h)

IRQSTATUS_RAW_SYS is shown in Figure 6-105 and described in Table 6-119.

Figure 6-105. IRQSTATUS_RAW_SYS Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-119. IRQSTATUS_RAW_SYS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	Reserved	R	0h	
2	reg_dnv_sys	R/W	0h	Raw status of system OCP interrupt for LPDDR2 NVM data not valid. Write 1 to set the (raw) status, mostly for debug. Writing a 0 has no effect.
1	reg_ta_sys	R/W	0h	Raw status of system OCP interrupt. Write 1 to set the (raw) status, mostly for debug. Writing a 0 has no effect.
0	reg_err_sys	R/W	0h	Raw status of system OCP interrupt. Write 1 to set the (raw) status, mostly for debug. Writing a 0 has no effect.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.18 IRQSTATUS_RAW_LL Register (offset = A8h)

IRQSTATUS_RAW_LL is shown in Figure 6-106 and described in Table 6-120.

Figure 6-106. IRQSTATUS_RAW_LL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-120. IRQSTATUS_RAW_LL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	Reserved	R	0h	
2	reg_dnv_ll	R/W	0h	Raw status of low-latency OCP interrupt for LPDDR2 NVM data not valid. Write 1 to set the (raw) status, mostly for debug. Writing a 0 has no effect.
1	reg_ta_ll	R/W	0h	Raw status of low-latency OCP interrupt. Write 1 to set the (raw) status, mostly for debug. Writing a 0 has no effect.
0	reg_err_ll	R/W	0h	Raw status of low-latency OCP interrupt. Write 1 to set the (raw) status, mostly for debug. Writing a 0 has no effect.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.19 IRQSTATUS_SYS Register (offset = ACh)

IRQSTATUS_SYS is shown in Figure 6-107 and described in Table 6-121.

Figure 6-107. IRQSTATUS_SYS Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-121. IRQSTATUS_SYS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	Reserved	R	0h	
2	reg_dnv_sys	R/W	0h	Enabled status of system OCP interrupt for LPDDR2 NVM data not valid. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, i.e. even if not enabled). Writing a 0 has no effect.
1	reg_ta_sys	R/W	0h	Enabled status of system OCP interrupt. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, i.e. even if not enabled). Writing a 0 has no effect.
0	reg_err_sys	R/W	0h	Enabled status of system OCP interrupt. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, i.e. even if not enabled). Writing a 0 has no effect.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.20 IRQSTATUS_LL Register (offset = B0h)

IRQSTATUS_LL is shown in Figure 6-108 and described in Table 6-122.

Figure 6-108. IRQSTATUS_LL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-122. IRQSTATUS_LL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	Reserved	R	0h	
2	reg_dnv_ll	R/W	0h	Enabled status of low-latency OCP interrupt for LPDDR2 NVM data not valid. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, i.e. even if not enabled). Writing a 0 has no effect.
1	reg_ta_ll	R/W	0h	Enabled status of low-latency OCP interrupt. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, i.e. even if not enabled). Writing a 0 has no effect.
0	reg_err_ll	R/W	0h	Enabled status of low-latency OCP interrupt. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, i.e. even if not enabled). Writing a 0 has no effect.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.21 IRQENABLE_SET_SYS Register (offset = B4h)

IRQENABLE_SET_SYS is shown in Figure 6-109 and described in Table 6-123.

Figure 6-109. IRQENABLE_SET_SYS Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-123. IRQENABLE_SET_SYS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	Reserved	R	0h	
2	reg_en_dnv_sys	R/W	0h	Enable set for system OCP interrupt for LPDDR2 NVM data not valid. Writing a 1 will enable the interrupt, and set this bit as well as the corresponding Interrupt Enable Clear Register. Writing a 0 has no effect.
1	reg_en_ta_sys	R/W	0h	Enable set for system OCP interrupt. Writing a 1 will enable the interrupt, and set this bit as well as the corresponding Interrupt Enable Clear Register. Writing a 0 has no effect.
0	reg_en_err_sys	R/W	0h	Enable set for system OCP interrupt. Writing a 1 will enable the interrupt, and set this bit as well as the corresponding Interrupt Enable Clear Register. Writing a 0 has no effect.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.22 IRQENABLE_SET_LL Register (offset = B8h)

IRQENABLE_SET_LL is shown in Figure 6-110 and described in Table 6-124.

Figure 6-110. IRQENABLE_SET_LL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-124. IRQENABLE_SET_LL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	Reserved	R	0h	
2	reg_en_dnv_ll	R/W	0h	Enable set for low-latency OCP interrupt for LPDDR2 NVM data not valid. Writing a 1 will enable the interrupt, and set this bit as well as the corresponding Interrupt Enable Clear Register. Writing a 0 has no effect.
1	reg_en_ta_ll	R/W	0h	Enable set for low-latency OCP interrupt. Writing a 1 will enable the interrupt, and set this bit as well as the corresponding Interrupt Enable Clear Register. Writing a 0 has no effect.
0	reg_en_err_ll	R/W	0h	Enable set for low-latency OCP interrupt. Writing a 1 will enable the interrupt, and set this bit as well as the corresponding Interrupt Enable Clear Register. Writing a 0 has no effect.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.23 IRQENABLE_CLR_SYS Register (offset = BCh)

IRQENABLE_CLR_SYS is shown in Figure 6-111 and described in Table 6-125.

Figure 6-111. IRQENABLE_CLR_SYS Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-125. IRQENABLE_CLR_SYS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	Reserved	R	0h	
2	reg_en_dnv_sys	R/W	0h	Enable clear for system OCP interrupt for LPDDR2 NVM data not valid. Writing a 1 will disable the interrupt, and clear this bit as well as the corresponding Interrupt Enable Set Register. Writing a 0 has no effect.
1	reg_en_ta_sys	R/W	0h	Enable clear for system OCP interrupt. Writing a 1 will disable the interrupt, and clear this bit as well as the corresponding Interrupt Enable Set Register. Writing a 0 has no effect.
0	reg_en_err_sys	R/W	0h	Enable clear for system OCP interrupt. Writing a 1 will disable the interrupt, and clear this bit as well as the corresponding Interrupt Enable Set Register. Writing a 0 has no effect.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.24 IRQENABLE_CLR_LL Register (offset = C0h)

IRQENABLE_CLR_LL is shown in Figure 6-112 and described in Table 6-126.

Figure 6-112. IRQENABLE_CLR_LL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-126. IRQENABLE_CLR_LL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	Reserved	R	0h	
2	reg_en_dnv_ll	R/W	0h	Enable clear for low-latency OCP interrupt for LPDDR2 NVM data not valid. Writing a 1 will disable the interrupt, and clear this bit as well as the corresponding Interrupt Enable Set Register. Writing a 0 has no effect.
1	reg_en_ta_ll	R/W	0h	Enable clear for low-latency OCP interrupt. Writing a 1 will disable the interrupt, and clear this bit as well as the corresponding Interrupt Enable Set Register. Writing a 0 has no effect.
0	reg_en_err_ll	R/W	0h	Enable clear for low-latency OCP interrupt. Writing a 1 will disable the interrupt, and clear this bit as well as the corresponding Interrupt Enable Set Register. Writing a 0 has no effect.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.25 SDRAM_REF_CTRL_SHDW Register (offset = 14h)

SDRAM_REF_CTRL_SHDW is shown in Figure 6-113 and described in Table 6-127.

Figure 6-113. SDRAM_REF_CTRL_SHDW Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-127. SDRAM_REF_CTRL_SHDW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	Reserved	R	0h	
15-0	reg_refresh_rate_shdw	R/W	0h	Shadow field for reg_refresh_rate. This field is loaded into reg_refresh_rate field in SDRAM Refresh Control register when SIdleAck is asserted. This register is not auto corrected when the value is invalid.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.26 ZQ_CONFIG Register (offset = C8h)

ZQ_CONFIG is shown in Figure 6-114 and described in Table 6-128.

Figure 6-114. ZQ_CONFIG Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-128. ZQ_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31	reg_zq_cs1en	R/W	0h	Writing a 1 enables ZQ calibration for CS1.
30	reg_zq_cs0en	R/W	0h	Writing a 1 enables ZQ calibration for CS0.
29	reg_zq_dualcalen	R/W	0h	ZQ Dual Calibration enable. Allows both ranks to be ZQ calibrated simultaneously. Setting this bit requires both chip selects to have a seerate calibration resistor per device.
28	reg_zq_sfexiten	R/W	0h	ZQCL on Self Refresh, Active Power-Down, and Precharge Power-Down exit enable. Writing a 1 enables the issuing of ZQCL on Self-Refresh, Active Power-Down, and Precharge Power-Down exit.
27-20	Reserved	R	0h	
19-18	reg_zq_zqinit_mult	R/W	0h	Indicates the number of ZQCL intervals that make up a ZQINIT interval, minus one.
17-16	reg_zq_zqcl_mult	R/W	0h	Indicates the number of ZQCS intervals that make up a ZQCL interval, minus one. ZQCS interval is defined by reg_zq_zqcs in SDRAM Timing 3 Register.
15-0	reg_zq_refinterval	R/W	0h	Number of refresh periods between ZQCS commans. This field supports between one refresh period to 256 ms between ZQCS calibration commands. Refresh period is defined by reg_refresh_rate in SDRAM Refresh Control register.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.27 TEMP_ALERT_CONFIG Register (offset = CCh)

TEMP_ALERT_CONFIG is shown in Figure 6-115 and described in Table 6-129.

Figure 6-115. TEMP_ALERT_CONFIG Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-129. TEMP_ALERT_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31	reg_ta_cs1en	R/W	0h	Writing a 1 enables temperature alert polling for CS1.
30	reg_ta_cs0en	R/W	0h	Writing a 1 enables temperature alert polling for CS0.
29	Reserved	R	0h	
28	reg_ta_sfexiten	R/W	0h	Temperature Alert Poll on Self-Refresh, Active Power-Down, and Precharge Power-Down exit enable. Writing a 1 enables the issuing of a temperature alert poll on Self-Refresh exit.
27-26	reg_ta_devwdt	R/W	0h	
25-24	reg_ta_devcnt	R/W	0h	
23-22	Reserved	R	0h	
21-0	reg_ta_refinterval	R/W	0h	Number of refresh periods between temperature alert polls. This field supports one refresh period to 10 seconds between temperature alert polls. Refresh period is defined by reg_refresh_rate in SDRAM Refresh Control register.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.28 OCP_ERR_LOG Register (offset = D0h)

OCP_ERR_LOG is shown in Figure 6-116 and described in Table 6-130.

Figure 6-116. OCP_ERR_LOG Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 6-130. OCP_ERR_LOG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	Reserved	R	0h	
15-14	reg_MAddrSpace	R	0h	Address space of the first errored transaction. Writing a 1 to the It field in the Interrupt Raw register or the It_masked field in the Interrupt Masked register will clear this field.
13-11	reg_MBurstSeq	R	0h	Addressing mode of the first errored transaction. Writing a 1 to the It field in the Interrupt Raw register or the It_masked field in the Interrupt Masked register will clear this field.
10-8	reg_MCmd	R	0h	Command type of the first errored transaction. Writing a 1 to the It field in the Interrupt Raw register or the It_masked field in the Interrupt Masked register will clear this field.
7-0	reg_MConnID	R	0h	Connection ID of the first errored transaction. Writing a 1 to the It field in the Interrupt Raw register or the It_masked field in the Interrupt Masked register will clear this field.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.29 DDR_PHY_CTRL_1 Register (offset = E4h)

DDR_PHY_CTRL_1 is shown in Figure 6-117 and described in Table 6-131.

Figure 6-117. DDR_PHY_CTRL_1 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-131. DDR_PHY_CTRL_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	reg_ddr_phy_ctrl_1	R/W	0h	This register is used to control the DDR PHY. The bit field definitions are DDR PHY specific and should be described in the DDR PHY specification.
3-0	reg_read_latency	R/W	0h	This field defines the read latency for the read data from SDRAM in number of m_clk cycles. This field is used by the EMIF as well as the PHY. The EMIF will expect the first read data to arrive (reg_read_latency + 3) m_clk cycles from the read command. Refer to the PHY specification for information on programming this field.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.30 DDR_PHY_CTRL_1_SHDW Register (offset = E8h)

DDR_PHY_CTRL_1_SHDW is shown in Figure 6-118 and described in Table 6-132.

Figure 6-118. DDR_PHY_CTRL_1_SHDW Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-132. DDR_PHY_CTRL_1_SHDW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	reg_ddr_phy_ctrl_1_shdw	R/W	0h	This register is used to control the DDR PHY. The bit field definitions are DDR PHY specific and should be described in the DDR PHY specification.
3-0	reg_read_latency_shdw	R/W	0h	Shadow field for read_latency. This field is loaded into reg_refresh_rate field in DDR PHY Control 1 register when SIdleAck is asserted.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.31 DDR_PHY_CTRL_2 Register (offset = ECh)

DDR_PHY_CTRL_2 is shown in Figure 6-119 and described in Table 6-133.

Figure 6-119. DDR_PHY_CTRL_2 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-133. DDR_PHY_CTRL_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	reg_ddr_phy_ctrl_2	R/W	0h	This register is used to control the DDR PHY. The bit field definitions are DDR PHY specific and should be described in the DDR PHY specification.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.32 SDRAM_TIM_1 Register (offset = 18h)

SDRAM_TIM_1 is shown in Figure 6-120 and described in Table 6-134.

Figure 6-120. SDRAM_TIM_1 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-134. SDRAM_TIM_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	Reserved	R	0h	
28-25	reg_t_rp	R/W	0h	Minimum number of m_clk cycles from Precharge to Activate or Refresh, minus one.
24-21	reg_t_rcd	R/W	0h	Minimum number of m_clk cycles from Activate to Read or Write, minus one.
20-17	reg_t_wr	R/W	0h	Minimum number of m_clk cycles from last Write transfer to Pre-charge, minus one. The SDRAM initialization sequence will be started when the value of this field is changed from the previous value and the EMIF is in DDR2 mode.
16-12	reg_t_ras	R/W	0h	Minimum number of m_clk cycles from Activate to Pre-charge, minus one. reg_t_ras >= reg_t_rcd.
11-6	reg_t_rc	R/W	0h	Minimum number of m_clk cycles from Activate to Activate, minus one.
5-3	reg_t_rrd	R/W	0h	Minimum number of m_clk cycles from Activate to Activate for a different bank, minus one. For an 8-bank DDR2, this field must be equal to ((((4*tRRD)+(2*tCK))/(4*tCK))-1).
2-0	reg_t_wtr	R/W	0h	Minimum number of m_clk cycles from last Write to Read, minus one.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.33 SDRAM_TIM_1_SHDW Register (offset = 1Ch)

SDRAM_TIM_1_SHDW is shown in Figure 6-121 and described in Table 6-135.

Figure 6-121. SDRAM_TIM_1_SHDW Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-135. SDRAM_TIM_1_SHDW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	Reserved	R	0h	
28-25	reg_t_rp_shdw	R/W	0h	Shadow field for reg_t_rp. This field is loaded into reg_t_rp field in SDRAM Timing 1 register when SIdleAck is asserted.
24-21	reg_t_rcd_shdw	R/W	0h	Shadow field for reg_t_rcd. This field is loaded into reg_t_rcd field in SDRAM Timing 1 register when SIdleAck is asserted.
20-17	reg_t_wr_shdw	R/W	0h	Shadow field for reg_t_wr. This field is loaded into reg_t_wr field in SDRAM Timing 1 register when SIdleAck is asserted. initialization sequence will be started when the value of this field is changed from the previous value and the EMIF is in DDR2 mode.
16-12	reg_t_ras_shdw	R/W	0h	Shadow field for reg_t_ras. This field is loaded into reg_t_ras field in SDRAM Timing 1 register when SIdleAck is asserted.
11-6	reg_t_rc_shdw	R/W	0h	Shadow field for reg_t_rc. This field is loaded into reg_t_rc field in SDRAM Timing 1 register when SIdleAck is asserted.
5-3	reg_t_rrd_shdw	R/W	0h	Shadow field for reg_t_rrd. This field is loaded into reg_t_rrd field in SDRAM Timing 1 register when SIdleAck is asserted.
2-0	reg_t_wtr_shdw	R/W	0h	Shadow field for reg_t_wtr. This field is loaded into reg_t_wtr field in SDRAM Timing 1 register when SIdleAck is asserted.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.34 SDRAM_TIM_2 Register (offset = 20h)

SDRAM_TIM_2 is shown in Figure 6-122 and described in Table 6-136.

Figure 6-122. SDRAM_TIM_2 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-136. SDRAM_TIM_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31	Reserved	R	0h	
30-28	reg_t_xp	R/W	0h	Minimum number of m_clk cycles from Powerdown exit to any command other than a Read command, minus one. For DDR2 and LPDDR1, this field must satisfy greater of tXP or tCKE.
27-25	reg_t_odt	R/W	0h	Minimum number of m_clk cycles from ODT enable to write data driven for DDR2 and DDR3. reg_t_odt must be equal to tAOND.
24-16	reg_t_xsnr	R/W	0h	Minimum number of m_clk cycles from Self-Refresh exit to any command other than a Read command, minus one.
15-6	reg_t_xsrd	R/W	0h	Minimum number of m_clk cycles from Self-Refresh exit to a Read command, minus one.
5-3	reg_t_rtp	R/W	0h	Minimum number of m_clk cycles from the last Read command to a Pre-charge command for DDR2, minus one.
2-0	reg_t_cke	R/W	0h	Minimum number of m_clk cycles between pad_cke_o changes, minus one.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.35 SDRAM_TIM_2_SHDW Register (offset = 24h)

SDRAM_TIM_2_SHDW is shown in Figure 6-123 and described in Table 6-137.

Figure 6-123. SDRAM_TIM_2_SHDW Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-137. SDRAM_TIM_2_SHDW Register Field Descriptions

Bit	Field	Type	Reset	Description
31	Reserved	R	0h	
30-28	reg_t_xp_shdw	R/W	0h	Shadow field for reg_t_xp. This field is loaded into reg_t_xp field in SDRAM Timing 2 register when SIdleAck is asserted.
27-25	reg_t_odt_shdw	R/W	0h	Shadow field for reg_t_odt. This field is loaded into reg_t_odt field in SDRAM Timing 2 register when SIdleAck is asserted.
24-16	reg_t_xsnr_shdw	R/W	0h	Shadow field for reg_t_xsnr. This field is loaded into reg_t_xsnr field in SDRAM Timing 2 register when SIdleAck is asserted.
15-6	reg_t_xsrd_shdw	R/W	0h	Shadow field for reg_t_xsrd. This field is loaded into reg_t_xsrd field in SDRAM Timing 2 register when SIdleAck is asserted.
5-3	reg_t_rtp_shdw	R/W	0h	Shadow field for reg_t_rtp. This field is loaded into reg_t_rtp field in SDRAM Timing 2 register when SIdleAck is asserted.
2-0	reg_t_cke_shdw	R/W	0h	Shadow field for reg_t_cke. This field is loaded into reg_t_cke field in SDRAM Timing 2 register when SIdleAck is asserted.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.36 STATUS Register (offset = 4h)

STATUS is shown in Figure 6-124 and described in Table 6-138.

Figure 6-124. STATUS Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-138. STATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31	reg_be	R	0h	Big Endian. Reflects the value on the config_big_endian port that defines whether the EMIF is in big or little endian mode.
30	reg_dual_clk_mode	R	0h	Dual Clock mode. Reflects the value on the config_dual_clk_mode port that defines whether the ocp_clk and m_clk are asynchronous.
29	reg_fast_init	R	0h	Fast Init. Reflects the value on the config_fast_init port that defines whether the EMIF fast initialization mode has been enabled.
28-3	Reserved	R	0h	
2	reg_phy_dll_ready	R	0h	DDR PHY Ready. Reflects the value on the phy_ready port (active high) that defines whether the DDR PHY is ready for normal operation.
1-0	Reserved	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.37 SDRAM_TIM_3 Register (offset = 28h)

SDRAM_TIM_3 is shown in Figure 6-125 and described in Table 6-139.

Figure 6-125. SDRAM_TIM_3 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-139. SDRAM_TIM_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	Reserved	R	0h	
23-21	reg_t_ckesr	R/W	0h	Minimum number of m_clk cycles for which LPDDR2 must remain in Self Refresh, minus one.
20-15	reg_zq_zqcs	R/W	0h	Number of m_clk clock cycles for a ZQCS command, minus one.
14-13	reg_t_tdqsckmax	R/W	0h	Number of m_clk that satisfies tDQSCKmax for LPDDR2, minus one.
12-4	reg_t_rfc	R/W	0h	Minimum number of m_clk cycles from Refresh or Load Mode to Refresh or Activate, minus one.
3-0	reg_t_ras_max	R/W	0h	Maximum number of reg_refresh_rate intervals from Activate to Precharge command. This field must be equal to ((tRASmax / tREFI)-1) rounded down to the next lower integer.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.38 SDRAM_TIM_3_SHDW Register (offset = 2Ch)

SDRAM_TIM_3_SHDW is shown in Figure 6-126 and described in Table 6-140.

Figure 6-126. SDRAM_TIM_3_SHDW Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-140. SDRAM_TIM_3_SHDW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	Reserved	R	0h	
23-21	reg_t_ckesr_shdw	R/W	0h	Shadow field for reg_t_ckesr. This field is loaded into reg_t_ckesr field in SDRAM Timing 3 register when SIdleAck is asserted.
20-15	reg_zq_zqcs_shdw	R/W	0h	Shadow field for reg_zq_zqcs. This field is loaded into reg_zq_zqcs field in SDRAM Timing 3 register when SIdleAck is asserted.
14-13	reg_t_tdqsckmax_shdw	R/W	0h	Shadow field for reg_t_dqsckmax. This field is loaded into reg_t_dqsckmax field in SDRAM Timing 3 register when SIdleAck is asserted.
12-4	reg_t_rfc_shdw	R/W	0h	Shadow field for reg_t_rfc. This field is loaded into reg_t_rfc field in SDRAM Timing 3 register when SIdleAck is asserted.
3-0	reg_t_ras_max_shdw	R/W	0h	Shadow field for reg_t_ras_max. This field is loaded into reg_t_ras_max field in SDRAM Timing 3 register when SIdleAck is asserted.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.39 LPDDR2_NVM_TIM Register (offset = 30h)

LPDDR2_NVM_TIM is shown in Figure 6-127 and described in Table 6-141.

Figure 6-127. LPDDR2_NVM_TIM Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-141. LPDDR2_NVM_TIM Register Field Descriptions

Bit	Field	Type	Reset	Description
31	Reserved	R	0h	
30-28	reg_nvm_t_xp	R/W	0h	Minimum number of m_clk cycles from Powerdown exit to any command, minus one.
27	Reserved	R	0h	
26-24	reg_nvm_t_wtr	R/W	0h	Minimum number of m_clk cycles from last Write to Read, minus one.
23-20	reg_nvm_t_rp	R/W	0h	Minimum number of m_clk cycles from Preactive to Activate, minus one.
19-16	reg_nvm_t_wra	R/W	0h	Minimum number of m_clk cycles from last Write transfer to Activate, minus one.
15-8	reg_nvm_t_rrd	R/W	0h	Minimum number of m_clk cycles from Activate to Activate for a different bank, minus one.
7-0	reg_nvm_t_rcdmin	R/W	0h	Minimum number of m_clk cycles from Activate to Read or Write, minus one.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.40 LPDDR2_NVM_TIM_SHDW Register (offset = 34h)

LPDDR2_NVM_TIM_SHDW is shown in Figure 6-128 and described in Table 6-142.

Figure 6-128. LPDDR2_NVM_TIM_SHDW Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-142. LPDDR2_NVM_TIM_SHDW Register Field Descriptions

Bit	Field	Type	Reset	Description
31	Reserved	R	0h	
30-28	reg_nvm_t_xp_shdw	R/W	0h	Shadow field for reg_nvm_t_xp. This field is loaded into reg_nvm_t_xp field in LPDDR2-NVM Timing register when SIdleAck is asserted.
27	Reserved	R	0h	
26-24	reg_nvm_t_wtr_shdw	R/W	0h	Shadow field for reg_nvm_t_wtr. This field is loaded into reg_nvm_t_wtr field in LPDDR2-NVM Timing register when SIdleAck is asserted.
23-20	reg_nvm_t_rp_shdw	R/W	0h	Shadow field for reg_nvm_t_rp. This field is loaded into reg_nvm_t_rp field in LPDDR2-NVM Timing register when SIdleAck is asserted.
19-16	reg_nvm_t_wra_shdw	R/W	0h	Shadow field for reg_nvm_t_wra. This field is loaded into reg_nvm_t_wra field in LPDDR2-NVM Timing register when SIdleAck is asserted.
15-8	reg_nvm_t_rrd_shdw	R/W	0h	Shadow field for reg_nvm_t_rrd. This field is loaded into reg_nvm_t_rrd field in LPDDR2-NVM Timing register when SIdleAck is asserted.
7-0	reg_nvm_t_rcdmin_shdw	R/W	0h	Shadow field for reg_nvm_t_rcdmin. This field is loaded into reg_nvm_t_rcdmin field in LPDDR2-NVM Timing register when SIdleAck is asserted.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.41 PWR_MGMT_CTRL Register (offset = 38h)

PWR_MGMT_CTRL is shown in Figure 6-129 and described in Table 6-143.

Figure 6-129. PWR_MGMT_CTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-143. PWR_MGMT_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	Reserved	R	0h	
15-12	reg_pd_tim	R/W	0h	Power Mangement timer for Power-Down. The EMIF will put the external SDRAM in Power-Down mode after the EMIF is idle for these number of m_clk cycles and if reg_lp_mode field is set to 4. Set to 0 to immediately enter Power-Down mode. Set to 1 for 16 clocks, set to 2 for 32 clocks, set to 3 for 64 clocks, set to 4 for 128 clocks, set to 5 for 256 clocks, set to 6 for 512 clocks, set to 7 for 1024 clocks, set to 8 for 2048 clocks, set to 9 for 4096 clocks, set to 10 for 8192 clocks, set to 11 for 16384 clocks, set to 12 for 32768 clocks, set to 13 for 65536 clocks, set to 14 for 131072 clocks, and set to 15 for 262144 clocks.
11	reg_dpd_en	R/W	0h	Deep Power Down enable. Set to 0 for normal operation. Set to 1 to enter deep power down mode. This mode will override the reg_lp_mode field setting.
10-8	reg_lp_mode	R/W	0h	Automatic Power Management enable. Set to 1 for Clock Stop, set to 2 for Self Refresh, and set to 4 for Power-Down. All other values will disable automatic power management.
7-4	reg_sr_tim	R/W	0h	Power Mangement timer for Self Refresh. The EMIF will put the external SDRAM in Self Refresh mode after the EMIF is idle for these number of m_clk cycles and if reg_lp_mode field is set to 2. Set to 0 to immediately enter Self Refresh mode. Set to 1 for 16 clocks, set to 2 for 32 clocks, set to 3 for 64 clocks, set to 4 for 128 clocks, set to 5 for 256 clocks, set to 6 for 512 clocks, set to 7 for 1024 clocks, set to 8 for 2048 clocks, set to 9 for 4096 clocks, set to 10 for 8192 clocks, set to 11 for 16384 clocks, set to 12 for 32768 clocks, set to 13 for 65536 clocks, set to 14 for 131072 clocks, and set to 15 for 262144 clocks.
3-0	reg_cs_tim	R/W	0h	Power Mangement timer for Clock Stop. The EMIF will put the external SDRAM in Clock Stop mode after the EMIF is idle for these number of m_clk cycles and if reg_lp_mode field is set to 1. Set to 0 to immediately enter Clock Stop mode. Set to 1 for 16 clocks, set to 2 for 32 clocks, set to 3 for 64 clocks, set to 4 for 128 clocks, set to 5 for 256 clocks, set to 6 for 512 clocks, set to 7 for 1024 clocks, set to 8 for 2048 clocks, set to 9 for 4096 clocks, set to 10 for 8192 clocks, set to 11 for 16384 clocks, set to 12 for 32768 clocks, set to 13 for 65536 clocks, set to 14 for 131072 clocks, and set to 15 for 262144 clocks.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.42 PWR_MGMT_CTRL_SHDW Register (offset = 3Ch)

PWR_MGMT_CTRL_SHDW is shown in Figure 6-130 and described in Table 6-144.

Figure 6-130. PWR_MGMT_CTRL_SHDW Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-144. PWR_MGMT_CTRL_SHDW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-12	Reserved	R	0h	
11-8	reg_pd_tim_shdw	R/W	0h	Shadow field for reg_pd_tim. This field is loaded into reg_pd_tim field in Power Management Control register when SIdleAck is asserted.
7-4	reg_sr_tim_shdw	R/W	0h	Shadow field for reg_sr_tim. This field is loaded into reg_sr_tim field in Power Management Control register when SIdleAck is asserted.
3-0	reg_cs_tim_shdw	R/W	0h	Shadow field for reg_cs_tim. This field is loaded into reg_cs_tim field in Power Management Control register when SIdleAck is asserted.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.43 LPDDR2_MODE_REG_DATA Register (offset = 40h)

LPDDR2_MODE_REG_DATA is shown in Figure 6-131 and described in Table 6-145.

Figure 6-131. LPDDR2_MODE_REG_DATA Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-145. LPDDR2_MODE_REG_DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	Reserved	R	0h	
6-0	reg_value_0	R/W	0h	Mode register value.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.44 SDRAM_CONFIG Register (offset = 8h)

SDRAM_CONFIG is shown in Figure 6-132 and described in Table 6-146.

Figure 6-132. SDRAM_CONFIG Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-146. SDRAM_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	reg_sdram_type	R/W	0h	SDRAM Type selection. Set to 0 for DDR1, set to 1 for LPDDR1, set to 2 for DDR2, set to 3 for DDR3, and set to 4 for LPDDR2. All other values are reserved.
28-27	reg_ibank_pos	R/W	0h	Internal bank position. Set to 0 to assign internal bank address bits from lower OCP address bits as shown in tables. Set to 1, 2, or 3 to assign internal bank address bits from higher OCP address as shown in tables.
26-24	reg_ddr_term	R/W	0h	DDR2 and DDR3 termination resistor value. Set to 0 to disable termination. For DDR2, set to 1 for 75 ohm, set to 2 for 150 ohm, and set to 3 for 50 ohm. For DDR3, set to 1 for RZQ/4, set to 2 for RZQ/2, set to 3 for RZQ/6, set to 4 for RZQ/12, and set to 5 for RZQ/8. All other values are reserved.
23	reg_ddr2_ddqs	R/W	0h	DDR2 and LPDDR2 differential DQS enable. Set to 0 for single ended DQS. Set to 1 for differential DQS.
22-21	reg_dyn_odt	R/W	0h	DDR3 Dynamic ODT. Set to 0 to turn off dynamic ODT. Set to 1 for RZQ/4 and set to 2 for RZQ/2. All other values are reserved.
20	reg_ddr_disable_dll	R/W	0h	Disable DLL select. Set to 1 to disable DLL inside SDRAM.
19-18	reg_sdram_drive	R/W	0h	SDRAM drive strength. For DDR1/DDR2, set to 0 for normal, and set to 1 for weak drive strength. For DDR3, set to 0 for RZQ/6 and set to 1 for RZQ/7. For LPDDR1, set to 0 for full, set to 1 for 1/2, set to 2 for 1/4, and set to 3 for 1/8 drive strength. All other values are reserved.
17-16	reg_cwl	R/W	0h	DDR3 CAS Write latency. Value of 0, 1, 2, and 3 (CAS write latency of 5, 6, 7, and 8) are supported. Use the lowest value supported for best performance. All other values are reserved.
15-14	reg_narrow_mode	R/W	0h	SDRAM data bus width. Set to 0 for 32-bit and set to 1 for 16-bit. All other values are reserved.
13-10	reg_cl	R/W	0h	CAS Latency. The value of this field defines the CAS latency to be used when accessing connected SDRAM devices. Value of 2, 3, 5, and 6 (CAS latency of 2, 3, 1.5, and 2.5) are supported for DDR1. Value of 2, 3, 4, and 5 (CAS latency of 2, 3, 4, and 5) are supported for DDR2. Value of 2, 4, 6, 8, 10, 12, and 14 (CAS latency of 5, 6, 7, 8, 9, 10, and 11) are supported for DDR3. Value of 2 and 3 (CAS latency of 2 and 3) are supported for LPDDR1. Value of 3, 4, 5, 6, 7, and 8 (CAS latency of 3, 4, 5, 6, 7, and 8) are supported for LPDDR2-SDRAM. All other values are reserved.
9-7	reg_rowsize	R/W	0h	Row Size. Defines the number of row address bits of connected SDRAM devices. Set to 0 for 9 row bits, set to 1 for 10 row bits, set to 2 for 11 row bits, set to 3 for 12 row bits, set to 4 for 13 row bits, set to 5 for 14 row bits, set to 6 for 15 row bits, and set to 7 for 16 row bits. This field is only used when reg_ibank_pos field in SDRAM Config register is set to 1, 2, or 3, or reg_ebank_pos field in SDRAM Config_2 register is set to 1."
6-4	reg_ibank	R/W	0h	Internal Bank setup. Defines number of banks inside connected SDRAM devices. Set to 0 for 1 bank, set to 1 for 2 banks, set to 2 for 4 banks, and set to 3 for 8 banks. All other values are reserved.
3	reg_ebank	R/W	0h	External chip select setup. Defines whether SDRAM accesses will use 1 or 2 chip select lines. Set to 0 to use pad_cs_o_n[0] only. Set to 1 to use pad_cs_o_n[1:0]. This bit will automaticlly be set to 0 if reg_cs1nvmen field in the LPDDR2-NVM is set to 1.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 6-146. SDRAM_CONFIG Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2-0	reg_pagesize	R/W	0h	Page Size. Defines the internal page size of connected SDRAM devices. Set to 0 for 256-word page (8 column bits), set to 1 for 512-word page (9 column bits), set to 2 for 1024-word page (10 column bits), and set to 3 for 2048-word page (11 column bits). All other values are reserved.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.45 LPDDR2_MODE_REG_CFG Register (offset = 50h)

LPDDR2_MODE_REG_CFG is shown in Figure 6-133 and described in Table 6-147.

Figure 6-133. LPDDR2_MODE_REG_CFG Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-147. LPDDR2_MODE_REG_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31	reg_cs	R/W	0h	Chip select to issue mode register command. Set to 0 for CS0 and set to 1 for CS1.
30	reg_refresh_en	R/W	0h	Refresh Enable after MRW write. If a Mode Data register write occurs with this bit set to 1, the refresh operations will commence.
29-8	Reserved	R	0h	
7-0	reg_address	R/W	0h	Mode register address.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.46 OCP_CONFIG Register (offset = 54h)

OCP_CONFIG is shown in Figure 6-134 and described in Table 6-148.

Figure 6-134. OCP_CONFIG Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-148. OCP_CONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved	R	0h	
27-24	reg_sys_thresh_max	R/W	4h	System OCP Threshold Maximum. The number of commands the system interface can consume in the command FIFO. The value is used to determine when to stop future request, writing a zero will reserve no space for the associated interface. In the event the value is set to zero and a request is seen for that interface, the command FIFO will assume a value of 1. Since the low-latency interface has effectively a higher priority, the only way for the system interface to use all command FIFO entries is to set the reg_ll_thresh_max to zero.
23-20	Reserved	R	0h	
19-16	reg_ll_thresh_max	R/W	4h	Low-latency OCP Threshold Maximum. The number of commands the low latency interface can consume in the command FIFO. The value is used to determine when to stop future request, writing a zero will reserve no space for the associated interface. In the event the value is set to zero and a request is seen for that interface, the command FIFO will assume a value of 1.
15-8	Reserved	R	0h	
7-0	reg_pr_old_count	R/W	8h	Priority Raise Old Counter. Number of memory transfers after which the EMIF momentarily raises the priority of old commands in the OCP Command FIFO.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.47 OCP_CFG_VAL_1 Register (offset = 58h)

OCP_CFG_VAL_1 is shown in Figure 6-135 and described in Table 6-149.

Figure 6-135. OCP_CFG_VAL_1 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-149. OCP_CFG_VAL_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	reg_sys_bus_width	R	2h	System OCP data bus width for a particular configuration. 0 = 32-bit wide, 1 = 64-bit wide, 2 = 128-bit wide, and 3 = 256-bit wide
29-28	reg_ll_bus_width	R	0h	Low-latency OCP data bus width for a particular configuration. 0 = 32-bit wide, 1 = 64-bit wide, 2 = 128-bit wide, and 3 = 256-bit wide
27-16	Reserved	R	0h	
15-8	reg_wr_fifo_depth	R	8h	Write Data FIFO depth for a particular configuration.
7-0	reg_cmd_fifo_depth	R	8h	Command FIFO depth for a particular configuration.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.48 OCP_CFG_VAL_2 Register (offset = 5Ch)

OCP_CFG_VAL_2 is shown in Figure 6-136 and described in Table 6-150.

Figure 6-136. OCP_CFG_VAL_2 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-150. OCP_CFG_VAL_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	Reserved	R	0h	
23-16	reg_rreg_fifo_depth	R	0h	Register Read Data FIFO depth for a particular configuration.
15-8	reg_rsd_fifo_depth	R	0h	SDRAM Read Data FIFO depth for a particular configuration.
7-0	reg_rcmd_fifo_depth	R	0h	Read Command FIFO depth for a particular configuration.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.3.5.49 IODFT_TLGC Register (offset = 60h)

IODFT_TLGC is shown in Figure 6-137 and described in Table 6-151.

Figure 6-137. IODFT_TLGC Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 6-151. IODFT_TLGC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	reg_tlec	R/W	0h	IODFT Test Logic Execution Counter. Contains the number of cycle that the MISR signature will be accumulated. Upon the expiration of the counter the MISR capture will be turned off.
15	Reserved	R	0h	
14	reg_mt	R/W	0h	MISR on/off trigger command. 0x0 = inactive/no affect. 0x1 = MISR capture start on the first write or read command to the memory and continues to update the signature until reg_tlec expires
13	reg_act_cap_en	R/W	1h	Active cycles capture enable. If set to a 1 the MISRs and pattern generators will shift only during active cycles. If set to a 0 the MISRs and pattern generators will shift every clock cycle.
12	reg_opg_ld	R/W	0h	Load pattern generators' initial value. Set to 1 to load an initial value in the pattern generators from reg_tlec.
11	Reserved	R	0h	
10	reg_reset_phy	R/W	0h	Reset DDR PHY. Writing a 1 to this bit will reset the DDR PHY. This bit will self clear to 0.
9	Reserved	R	0h	
8	reg_mms	R/W	0h	Chooses the source of the MISR input. Set to 0 for output register, and set to 1 for input capture.
7-6	Reserved	R	0h	
5-4	reg_mc	R/W	0h	MISR state. Set to 0 to download results. Set to 1 to hold current value. Set to 2 to load initial value from pc bits. Set to 3 to enable MISR to capture signature.
3-1	reg_pc	R/W	0h	Pattern code. Defines the type of pattern that is selected for the pattern generators. Set to 0 for functional mode and set to 4 to hold current register value. For output pattern generator, set to 1 for random XOR, set to 2 for random XNOR, and set to 3 for an 8 bit shifter. For input pattern generator, set to 5 for random XOR, set to 6 for random XNOR, and set to 7 for an 8 bit shifter.
0	reg_tm	R/W	1h	Functional mode enable. Set to 1 for functional mode, and set to 0 for IODFT mode.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.4 ELM

6.4.1 Introduction

Non-managed NAND flash memories can be dense and nonvolatile in their own nature, but error-prone. When reading from NAND flash memories, some level of error-correction is required. In the case of NAND modules with no internal correction capability, sometimes referred to as bare NANDs, the correction process is delegated to the memory controller.

The general-purpose memory controller (GPMC) probes data read from an external NAND flash and uses this to compute checksum-like information, called syndrome polynomials, on a per-block basis. Each syndrome polynomial gives a status of the read operations for a full block, including 512 bytes of data, parity bits, and an optional spare-area data field, with a maximum block size of 1023 bytes. Computation is based on a Bose-ChaudhurI-Hocquenghem (BCH) algorithm. The error-location module (ELM) extracts error addresses from these syndrome polynomials.

Based on the syndrome polynomial value, the ELM can detect errors, compute the number of errors, and give the location of each error bit. The actual data is not required to complete the error-correction algorithm. Errors can be reported anywhere in the NAND flash block, including in the parity bits.

The maximum acceptable number of errors that can be corrected depends on a programmable configuration parameter. 4-, 8-, and 16-bit error-correction levels are supported. The ELM relies on a static and fixed definition of the generator polynomial for each error-correction level that corresponds to the generator polynomials defined in the GPMC (there are three fixed polynomial for the three correction error levels). A larger number of errors than the programmed error-correction level may be detected, but the ELM cannot correct them all. The offending block is then tagged as uncorrectable in the associated computation exit status register. If the computation is successful, that is, if the number of errors detected does not exceed the maximum value authorized for the chosen correction capability, the exit status register contains the information on the number of detected errors.

When the error-location process completes, an interrupt is triggered to inform the central processing unit (CPU) that its status can be checked. The number of detected errors and their locations in the NAND block can be retrieved from the module through register accesses.

6.4.1.1 ELM Features

The ELM has the following features:

- 4, 8, and 16 bits per 512-byte block error-location based on BCH algorithms
- Eight simultaneous processing contexts
- Page-based and continuous modes
- Interrupt generation on error-location process completion:
  - When the full page has been processed in page mode
  - For each syndrome polynomial in continuous mode

6.4.1.2 Unsupported ELM Features

There are no unsupported ELM features in this device.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.4.2 Integration

The error location module (ELM) is used to extract error addresses from syndrome polynomials generated using a BCH algorithm. Each of these polynomials gives a status of the read operations for a 512 bytes block from a NAND flash and its associated BCH parity bits, plus optionally some spare area information.

The ELM is intended to be used in conjunction with the GPMC. Syndrome polynomials generated on-the-fly when reading a NAND Flash page and stored in GPMC registers are passed to the ELM module. The MPU can then easily correct the data block by flipping the bits pointed to by the ELM error locations outputs.

6.4.2.1 ELM Connectivity Attributes

The general connectivity for the ELM module in this device is summarized in Table 6-152.

Table 6-152. ELM Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	L4PER_L4LS_GCLK
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	1 interrupt to Sabertooth
DMA Requests	None
Physical Address	L4 Peripheral slave port

6.4.2.2 ELM Clock and Reset Management

The ELM operates from a single OCP interface clock.

Table 6-153. ELM Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
Functional/interface clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_l4ls_gclk From PRCM

6.4.2.3 ELM Pin List

The ELM module does not include any external interface pins.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.4.3 Functional Description

The ELM is designed around the error-location engine, which handles the computation based on the input syndrome polynomials.

The ELM maps the error-location engine to a standard interconnect interface by using a set of registers to control inputs and outputs.

6.4.3.1 ELM Software Reset

To perform a software reset, write a 1 to the ELM_SYSCONFIG[1] SOFTRESET bit. The ELM_SYSSTATUS[0] RESETDONE bit indicates that the software reset is complete when its value is 1. When the software reset completes, the ELM_SYSCONFIG[1] SOFTRESET bit is automatically reset.

6.4.3.2 ELM Power Management

Table 6-154 describes the power-management features available to the ELM module.

Table 6-154. Local Power Management Features

Feature	Registers	Description
Clock autogating	ELM_SYSCONFIG[0] AUTOGATING bit	This bit allows a local power optimization inside the module by gating the ELM_FCLK clock upon the interface activity.
Slave idle modes	ELM_SYSCONFIG[4:3] SIDLEMODE bit field	Force-idle, No-idle, and Smart-idle modes are available.
Clock activity	ELM_SYSCONFIG[8] CLOCKACTIVITY bit	The clock can be switched-off or maintained during the wake-up period.
Master Standby modes	N/A	
Global Wake-up Enable	N/A	
Wake-up Sources Enable	N/A	

CAUTION

The PRCM module has no hardware means of reading CLOCKACTIVITY settings. Thus, software must ensure consistent programming between the ELM CLOCKACTIVITY and ELM clock PRCM control bits.

6.4.3.3 ELM Interrupt Requests

Table 6-155 lists the event flags, and their masks, that can cause module interrupts.

Table 6-155. Events

Event Flag	Event Mask	Map to	Description
ELM_IRQSTATUS[8] PAGE_VALID	ELM_IRQENABLE[8] PAGE_MASK	ELM_IRQ	Page interrupt
ELM_IRQSTATUS[7] LOC_VALID_7	ELM_IRQENABLE[7] LOCATION_MASK_7	ELM_IRQ	Error-location interrupt for syndrome polynomial 7
ELM_IRQSTATUS[6] LOC_VALID_6	ELM_IRQENABLE[6] LOCATION_MASK_6	ELM_IRQ	Error-location interrupt for syndrome polynomial 6
ELM_IRQSTATUS[5] LOC_VALID_5	ELM_IRQENABLE[5] LOCATION_MASK_5	ELM_IRQ	Error-location interrupt for syndrome polynomial 5
ELM_IRQSTATUS[4] LOC_VALID_4	ELM_IRQENABLE[4] LOCATION_MASK_4	ELM_IRQ	Error-location interrupt for syndrome polynomial 4
ELM_IRQSTATUS[3] LOC_VALID_3	ELM_IRQENABLE[3] LOCATION_MASK_3	ELM_IRQ	Error-location interrupt for syndrome polynomial 3
ELM_IRQSTATUS[2] LOC_VALID_2	ELM_IRQENABLE[2] LOCATION_MASK_2	ELM_IRQ	Error-location interrupt for syndrome polynomial 2

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 6-155. Events (continued)

Event Flag	Event Mask	Map to	Description
ELM_IRQSTATUS[1] LOC_VALID_1	ELM_IRQENABLE[1] LOCATION_MASK_1	ELM_IRQ	Error-location interrupt for syndrome polynomial 1
ELM_IRQSTATUS[0] LOC_VALID_0	ELM_IRQENABLE[0] LOCATION_MASK_0	ELM_IRQ	Error-location interrupt for syndrome polynomial 0

6.4.3.4 Processing Initialization

ELM_LOCATION_CONFIG global setting parameters must be set before using the error-location engine. The ELM_LOCATION_CONFIG[1:0] ECC_BCH_LEVEL bit defines the error-correction level used (4-,8-, or 16-bit error-correction). The ELM_LOCATION_CONFIG[26:16] ECC_SIZE bit field defines the maximum buffer length beyond which the engine processing no longer looks for errors.

The CPU can choose to use the ELM in continuous mode or page mode. If all ELM_PAGE_CTRL[i] SECTOR_i bits are reset (i is the syndrome polynomial number, i = 0 to 7), continuous mode is used. In any other case, page mode is implicitly selected.

- Continuous mode: Each syndrome polynomial is processed independently – results for a syndrome can be retrieved and acknowledged at any time, whatever the status of the other seven processing contexts.
- Page mode: Syndrome polynomials are grouped into atomic entities – only one page can be processed at any given time, even if all eight contexts are not used for this page. Unused contexts are lost and cannot be affected to any other processing. The full page must be acknowledged and cleared before moving to the next page.

For completion interrupts to be generated correctly, all ELM_IRQENABLE[i] LOCATION_MASK_i bits (i = 0 to 7) must be forced to 0 when in page mode, and set to 1 in continuous mode. Additionally, the ELM_IRQENABLE[8] PAGE_MASK bit must be set to 1 when in page mode.

The CPU initiates error-location processing by writing a syndrome polynomial into one of the eight possible register sets. Each of these register sets includes seven registers: ELM_SYNDROME_FRAGMENT_0_i to ELM_SYNDROME_FRAGMENT_6_i. The first six registers can be written in any order, but ELM_SYNDROME_FRAGMENT_6_i must be written last because it includes the validity bit, which instructs the ELM that this syndrome polynomial must be processed (the ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID bit).

As soon as one validity bit is asserted (ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID = 0x1, with i = 0 to 7), error-location processing can start for the corresponding syndrome polynomial. The associated ELM_LOCATION_STATUS_i and ELM_ERROR_LOCATION_0_i to ELM_ERROR_LOCATION_15_i registers are not reset (i = 0 to 7). The software must not consider them until the corresponding ELM_IRQSTATUS[i] LOC_VALID_i bit is set.

6.4.3.5 Processing Sequence

While the error-location engine is busy processing one syndrome polynomial, further syndrome polynomials can be written. They are processed when the current processing completes.

The engine completes early when:

- No error is detected; that is, when the ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE bit is set to 1 and the ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS bit field is set to 0x0.
- Too many errors are detected; that is, when the ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE bit is set to 0 while the ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS bit field is set with the value output by the error-location engine. The reported number of errors is not ensured if ECC_CORRECTABLE is 0.

If the engine completes early, the associated error-location registers ELM_ERROR_LOCATION_0_i to ELM_ERROR_LOCATION_15_i are not updated (i = 0 to 7).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

In all other cases, the engine goes through the entire error-location process. Each time an error-location is found, it is logged in the associated ECC_ERROR_LOCATION bit field. The first error detected is logged in the ELM_ERROR_LOCATION_0_i[12:0] ECC_ERROR_LOCATION bit field; the second in the ELM_ERROR_LOCATION_1_i[12:0] ECC_ERROR_LOCATION bit field, and so on.

Table 6-156. ELM_LOCATION_STATUS_i Value Decoding Table

ECC_CORRECTABLE Value	ECC_NB_ERRORS Value	Status	Number of Errors Detected	Action Required
1	0	OK	0	None
1	≠ 0	OK	ECC_NB_ERRORS	Correct the data buffer read based on the ELM_ERROR_LOCATION_0_i to ELM_ERROR_LOCATION_15_i results.
0	Any	Failed	Unknown	Software-dependant

6.4.3.6 Processing Completion

When the processing for a given syndrome polynomial completes, its ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID bit is reset. It must not be set again until the exit status registers, ELM_LOCATION_STATUS_i (i = 0 to 7), for this processing are checked. Failure to comply with this rule leads to potential loss of the first polynomial process data output.

The error-location engine signals the process completion to the ELM. When this event is detected, the corresponding ELM_IRQSTATUS[i] LOC_VALID_i bit is set (i = 0 to 7). The processing-exit status is available from the associated ELM_LOCATION_STATUS_i register, and error locations are stored in order in the ECC_ERROR_LOCATION fields. The software must only read valid error-location registers based on the number of errors detected and located.

Immediately after the error-location engine completes, a new syndrome polynomial can be processed, if any is available, as reported by the ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID validity bit, depending on the configured error-correction level. If several syndrome polynomials are available, a round-robin arbitration is used to select one for processing.

In continuous mode (that is, all bits in ELM_PAGE_CTRL are reset), an interrupt is triggered whenever a ELM_IRQSTATUS[i] LOC_VALID_i bit is asserted. The CPU must read the ELM_IRQSTATUS register to determine which polynomial is processed and retrieve the exit status and error locations (ELM_LOCATION_STATUS_i and ELM_ERROR_LOCATION_0_i to ELM_ERROR_LOCATION_15_i). When done, the CPU must clear the corresponding ELM_IRQSTATUS[i] LOC_VALID_i bit by writing it to 1. Other status bits must be written to 0 so that other interrupts are not unintentionally cleared. When using this mode, the ELM_IRQSTATUS[8] PAGE_VALID interrupt is never triggered.

In page mode, the module does not trigger interrupts for the processing completion of each polynomial because the ELM_IRQENABLE[i] LOCATION_MASK_i bits are cleared. A page is defined using the ELM_PAGE_CTRL register. Each SECTOR_i bit set means the corresponding polynomial i is part of the page processing. A page is fully processed when all tagged polynomials have been processed, as logged in the ELM_IRQSTATUS[i] LOC_VALID_i bit fields. The module triggers an ELM_IRQSTATUS[8] PAGE_VALID interrupt whenever it detects that the full page has been processed. To make sure the next page can be correctly processed, all status bits in the ELM_IRQSTATUS register must be cleared by using a single atomic-write access.

NOTE: Do not modify page setting parameters in the ELM_PAGE_CTRL register unless the engine is idle, no polynomial input is valid, and all interrupts have been cleared.

Because no polynomial-level interrupt is triggered in page mode, polynomials cleared in the ELM_PAGE_CTRL[i] SECTOR_i bit fields (i = 0 to 7) are processed as usual, but are essentially ignored. The CPU must manually poll the ELM_IRQSTATUS bits to check for their status.

Copyright © 2011, Texas Instruments Incorporated

6.4.4 Basic Programming Model

6.4.4.1 ELM Low Level Programming Model

6.4.4.1.1 Processing Initialization

Table 6-157. ELM Processing Initialization

Step	Register/ Bit Field / Programming Model	Value
Resets the module	ELM_SYSCONFIG[1] SOFTRESET	0x1
Wait until reset is done.	ELM_SYSSTATUS[0] RESETDONE	0x1
Configure the slave interface power management.	ELM_SYSCONFIG[4:3] SIDLEMODE	Set value
Defines the error-correction level used	ELM_LOCATION_CONFIG[1:0] ECC_BCH_LEVEL	Set value
Defines the maximum buffer length	ELM_LOCATION_CONFIG[26:16] ECC_SIZE	Set value
Sets the ELM in continuous mode or page mode	ELM_PAGE_CTRL	Set value
If continuous mode is used	All ELM_PAGE_CTRL[i] SECTOR_i (i = 0 to 7)	0x0
Enables interrupt for syndrome polynomial i	ELM_IRQENABLE[i] LOCATION_MASK_i	0x1
else (page mode is used)	One syndrome polynomial i is set ELM_PAGE_CTRL[i] SECTOR_i (i = 0 to 7)	0x1
Disable all interrupts for syndrome polynomial and enable PAGE_MASK interrupt.	All ELM_IRQENABLE[i] LOCATION_MASK_i = 0x0 and ELM_IRQENABLE[8] PAGE_MASK = 0x1	Set value
endif		Set value
Set the input syndrome polynomial i.	ELM_SYNDROME_FRAGMENT_0_i	Set value
	ELM_SYNDROME_FRAGMENT_1_i	Set value
	ELM_SYNDROME_FRAGMENT_5_i	Set value
	ELM_SYNDROME_FRAGMENT_6_i	Set value
Initiates the computation process	ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID	0x1

6.4.4.1.2 Read Results

The engine goes through the entire error-location process and results can be read. Table 6-158 and Table 6-159 describe the processing completion for continuous and page modes, respectively.

Table 6-158. ELM Processing Completion for Continuous Mode

Step	Register/ Bit Field / Programming Model	Value
Wait until process is complete for syndrome polynomial i: Wait until the ELM_IRQ interrupt is generated, or poll the status register.		
Read for which i the error-location process is complete.	ELM_IRQSTATUS[i] LOC_VALID_i	0x1
if the process fails (too many errors)	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE	0x0
It is software dependant.		
else (process successful, the engine completes)	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE	0x1
Read the number of errors.	ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS	
Read the error-location bit addresses for syndrome polynomial i of the ECC_NB_ERRORS first registers. It is the software responsibility to correct errors in the data buffer.	ELM_ERROR_LOCATION_0_i[12:0] ECC_ERROR_LOCATION	
	ELM_ERROR_LOCATION_1_i[12:0] ECC_ERROR_LOCATION	
	...	
	ELM_ERROR_LOCATION_15_i[12:0] ECC_ERROR_LOCATION	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 6-158. ELM Processing Completion for Continuous Mode (continued)

Step	Register/ Bit Field / Programming Model	Value
endif		
Clear the corresponding i interrupt.	ELM_IRQSTATUS[i] LOC_VALID_i	0x1

A new syndrome polynomial can be processed after the end of processing (ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID = 0x0) and after the exit status register check (ELM_LOCATION_STATUS_i).

Table 6-159. ELM Processing Completion for Page Mode

Step	Register/ Bit Field / Programming Model	Value
Wait until process is complete for syndrome polynomial i: Wait until the ELM_IRQ interrupt is generated, or poll the status register.		
Wait for page completed interrupt: All error locations are valid.	ELM_IRQSTATUS[8] PAGE_VALID	0x1
Repeat the following actions the necessary number of times. That is, once for each valid defined block in the page.		
Read the process exit status.	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE	
if the process fails (too many errors)	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE	0x0
It is software dependant.		
else (process successful, the engine completes)	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE	0x1
Read the number of errors.	ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS	
Read the error-location bit addresses for syndrome polynomial i of the ECC_NB_ERRORS first registers.	ELM_ERROR_LOCATION_0_i[12:0] ECC_ERROR_LOCATION	
	ELM_ERROR_LOCATION_1_i[12:0] ECC_ERROR_LOCATION	
	...	
	ELM_ERROR_LOCATION_15_i[12:0] ECC_ERROR_LOCATION	
endif		
End Repeat		
Clear the ELM_IRQSTATUS register.	ELM_IRQSTATUS	0x1FF

Next page can be correctly processed after a page is fully processed, when all tagged polynomials have been processed (ELM_IRQSTATUS[i] LOC_VALID_i = 0x1 for all syndrome polynomials i used in the page).

6.4.4.2 Use Case: ELM Used in Continuous Mode

In this example, the ELM module is programmed for an 8-bit error-correction capability in continuous mode. After reading a 528-byte NAND flash sector (512B data plus 16B spare area) with a 16-bit interface, a non-zero polynomial syndrome is reported from the GPMC (Polynomial syndrome 0 is used in the ELM):

- P = 0x0A16ABE115E44F767BFB0D0980

Table 6-160. Use Case: Continuous Mode

Step	Register/ Bit Field / Programming Model	Value
Resets the module	ELM_SYSCONFIG[1] SOFTRESET	0x1
Wait until reset is done.	ELM_SYSSTATUS[0] RESETDONE	0x1
Configure the slave interface power management: Smart idle is used.	ELM_SYSCONFIG[4:3] SIDLEMODE	0x2

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Table 6-160. Use Case: Continuous Mode (continued)

Step	Register/ Bit Field / Programming Model	Value
Defines the error-correction level used: 8 bits	ELM_LOCATION_CONFIG[1:0] ECC_BCH_LEVEL	0x1
Defines the maximum buffer length: 528 bytes (2x528 = 1056)	ELM_LOCATION_CONFIG[26:16] ECC_SIZE	0x420
Sets the ELM in continuous mode	ELM_PAGE_CTRL	0
Enables interrupt for syndrome polynomial 0	ELM_IRQENABLE[0] LOCATION_MASK_0	0x1
Set the input syndrome polynomial 0.	ELM_SYNDROME_FRAGMENT_0_i (i=0)	0xFB0D0980
	ELM_SYNDROME_FRAGMENT_1_i (i=0)	0xE44F767B
	ELM_SYNDROME_FRAGMENT_2_i (i=0)	0x16ABE115
	ELM_SYNDROME_FRAGMENT_3_i (i=0)	0x0000000A
Initiates the computation process	ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID (i=0)	0x1
Wait until process is complete for syndrome polynomial 0: IRQ_ELM is generated or poll the status register.		
Read that error-location process is complete for syndrome polynomial 0.	ELM_IRQSTATUS[0] LOC_VALID_0	0x1
Read the process exit status: All errors were successfully located.	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE (i=0)	0x1
Read the number of errors: Four errors detected.	ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS (i=0)	0x4
Read the error-location bit addresses for syndrome polynomial 0 of the 4 first registers: Errors are located in the data buffer at decimal addresses 431, 1062, 1909, 3452.	ELM_ERROR_LOCATION_0_i (i=0)	0x1AF
	ELM_ERROR_LOCATION_1_i (i=0)	0x426
	ELM_ERROR_LOCATION_2_i (i=0)	0x775
	ELM_ERROR_LOCATION_3_i (i=0)	0xD7C
Clear the corresponding interrupt for polynomial 0.	ELM_IRQSTATUS[0] LOC_VALID_0	0x1

The NAND flash data in the sector are seen as a polynomial of degree 4223 (number of bits in a 528 byte buffer minus 1), with each data bit being a coefficient in the polynomial. When reading from a NAND flash using the GPMC module, computation of the polynomial syndrome assumes that the first NAND word read at address 0x0 contains the highest-order coefficient in the message. Furthermore, in the 16-bit NAND word, bits are ordered from bit 7 to bit 0, then from bit 15 to bit 8. Based on this convention, an address table of the data buffer can be built. NAND memory addresses in Table 6-161 are given in decimal format.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 6-161. 16-bit NAND Sector Buffer Address Map

NAND Memory Address	Message bit addresses in the memory word															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	4215	4214	4213	4212	4211	4210	4209	4208	4223	4222	4221	4220	4219	4218	4217	4216
1	4175	4174	4173	4172	4171	4170	4169	4168	4183	4182	4181	4180	4179	4178	4177	4176
...																
47	3463	3462	3461	3460	3459	3458	3457	3456	3471	3470	3469	3468	3467	3466	3465	3464
48	3447	3446	3445	3444	3443	3442	3441	3440	3455	3454	3453	3452	3451	3450	3449	3448
49	3431	3430	3429	3428	3427	3426	3425	3424	3439	3438	3437	3436	3435	3434	3433	3432
50	3415	3414	3413	3412	3411	3410	3409	3408	3423	3422	3421	3420	3419	3418	3417	3416
...																
255	135	134	133	132	131	130	129	128	143	142	141	140	139	138	137	136
256	119	118	117	116	115	114	113	112	127	126	125	124	123	122	121	120
257	103	102	101	100	99	98	97	96	111	110	109	108	107	106	105	104
258	87	86	85	84	83	82	81	80	95	94	93	92	91	90	89	88
259	71	70	69	68	67	66	65	64	79	78	77	76	75	74	73	72
260	55	54	53	52	51	50	49	48	63	62	61	60	59	58	57	56
261	39	38	37	36	35	34	33	32	47	46	45	44	43	42	41	40
262	23	22	21	20	19	18	17	16	31	30	29	28	27	26	25	24
263	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8

The table can now be used to determine which bits in the buffer were incorrect and must be flipped. In this example, the first bit to be flipped is bit 4 from the 49th byte read from memory. It is up to the processor to correctly map this word to the copied buffer and to flip this bit. The same process must be repeated for all detected errors.

6.4.4.3 Use Case: ELM Used in Page Mode

In this example, the ELM module is programmed for an 16-bit error-correction capability in page mode. After reading a 528-byte NAND flash sector (512B data plus 16B spare area) with a 16-bit interface, four non-zero polynomial syndromes are reported from the GPMC (Polynomial syndrome 0, 1, 2, and 3 are used in the ELM):

- P0 = 0xE8B0 12ADDB5A318E05BE B0693DB28330B5CC A329AA05E0B718EF
- P1 = 0xBAD0 49A0D932C22E6669 0948DF08BE093336 79C6BA10E5F935EB
- P2 = 0x69D9 B86ABCD5EC3697FA A6498FEE54556EA0 1579EF7D60BA3189
- P3 = 0x0

Table 6-162. Use Case: Page Mode

Step	Register/ Bit Field / Programming Model	Value
Resets the module	ELM_SYSCONFIG[1] SOFTRESET	0x1
Wait until reset is done.	ELM_SYSSTATUS[0] RESETDONE	0x1
Configure the slave interface power management: Smart idle is used.	ELM_SYSCONFIG[4:3] SIDLEMODE	0x2
Defines the error-correction level used: 16 bits	ELM_LOCATION_CONFIG[1:0] ECC_BCH_LEVEL	0x2
Defines the maximum buffer length: 528 bytes	ELM_LOCATION_CONFIG[26:16] ECC_SIZE	0x420
Sets the ELM in page mode (4 blocks in a page)	ELM_PAGE_CTRL[0] SECTOR_0	0x1
	ELM_PAGE_CTRL[1] SECTOR_1	0x1
	ELM_PAGE_CTRL[2] SECTOR_2	0x1
	ELM_PAGE_CTRL[3] SECTOR_3	0x1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 6-162. Use Case: Page Mode (continued)

Step	Register/ Bit Field / Programming Model	Value
Disable all interrupts for syndrome polynomial and enable PAGE_MASK interrupt.	ELM_IRQENABLE	0x100
Set the input syndrome polynomial 0.	ELM_SYNDROME_FRAGMENT_0_i (i=0)	0xE0B718EF
	ELM_SYNDROME_FRAGMENT_1_i (i=0)	0xA329AA05
	ELM_SYNDROME_FRAGMENT_2_i (i=0)	0x8330B5CC
	ELM_SYNDROME_FRAGMENT_3_i (i=0)	0xB0693DB2
	ELM_SYNDROME_FRAGMENT_4_i (i=0)	0x318E05BE
	ELM_SYNDROME_FRAGMENT_5_i (i=0)	0x12ADDB5A
	ELM_SYNDROME_FRAGMENT_6_i (i=0)	0xE8B0
Set the input syndrome polynomial 1.	ELM_SYNDROME_FRAGMENT_0_i (i=1)	0xE5F935EB
	ELM_SYNDROME_FRAGMENT_1_i (i=1)	0x79C6BA10
	ELM_SYNDROME_FRAGMENT_2_i (i=1)	0xBE093336
	ELM_SYNDROME_FRAGMENT_3_i (i=1)	0x0948DF08
	ELM_SYNDROME_FRAGMENT_4_i (i=1)	0xC22E6669
	ELM_SYNDROME_FRAGMENT_5_i (i=1)	0x49A0D932
	ELM_SYNDROME_FRAGMENT_6_i (i=1)	0xBAD0
Set the input syndrome polynomial 2.	ELM_SYNDROME_FRAGMENT_0_i (i=2)	0x60BA3189
	ELM_SYNDROME_FRAGMENT_1_i (i=2)	0x1579EF7D
	ELM_SYNDROME_FRAGMENT_2_i (i=2)	0x54556EA0
	ELM_SYNDROME_FRAGMENT_3_i (i=2)	0xA6498FEE
	ELM_SYNDROME_FRAGMENT_4_i (i=2)	0xEC3697FA
	ELM_SYNDROME_FRAGMENT_5_i (i=2)	0xB86ABCD5
	ELM_SYNDROME_FRAGMENT_6_i (i=2)	0x69D9
Set the input syndrome polynomial 3.	ELM_SYNDROME_FRAGMENT_0_i (i=3)	0x0
	ELM_SYNDROME_FRAGMENT_1_i (i=3)	0x0
	ELM_SYNDROME_FRAGMENT_2_i (i=3)	0x0
	ELM_SYNDROME_FRAGMENT_3_i (i=3)	0x0
	ELM_SYNDROME_FRAGMENT_4_i (i=3)	0x0
	ELM_SYNDROME_FRAGMENT_5_i (i=3)	0x0
	ELM_SYNDROME_FRAGMENT_6_i (i=3)	0x0
Initiates the computation process for syndrome polynomial 0	ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID (i=0)	0x1
Initiates the computation process for syndrome polynomial 1	ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID (i=1)	0x1
Initiates the computation process for syndrome polynomial 2	ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID (i=2)	0x1
Initiates the computation process for syndrome polynomial 3	ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID (i=3)	0x1
Wait until process is complete for syndrome polynomial 0, 1, 2, and 3: Wait until the ELM_IRQ interrupt is generated or poll the status register.		
Wait for page completed interrupt: All error locations are valid.	ELM_IRQSTATUS[8] PAGE_VALID	0x1
Read the process exit status for syndrome polynomial 0: All errors were successfully located.	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE (i=0)	0x1
Read the process exit status for syndrome polynomial 1: All errors were successfully located.	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE (i=1)	0x1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 6-162. Use Case: Page Mode (continued)

Step	Register/ Bit Field / Programming Model	Value
Read the process exit status for syndrome polynomial 2: All errors were successfully located.	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE (i=2)	0x1
Read the process exit status for syndrome polynomial 3: All errors were successfully located.	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE (i=3)	0x1
Read the number of errors for syndrome polynomial 0: 4 errors detected.	ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS (i=0)	0x4
Read the number of errors for syndrome polynomial 1: 2 errors detected.	ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS (i=1)	0x2
Read the number of errors for syndrome polynomial 2: 1 error detected.	ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS (i=2)	0x1
Read the number of errors for syndrome polynomial 3: 0 errors detected.	ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS (i=3)	0x0
Read the error-location bit addresses for syndrome polynomial 0 of the 4 first registers:	ELM_ERROR_LOCATION_0_i (i=0)	0x1FE
	ELM_ERROR_LOCATION_1_i (i=0)	0x617
	ELM_ERROR_LOCATION_2_i (i=0)	0x650
	ELM_ERROR_LOCATION_3_i (i=0)	0xA83
Read the error-location bit addresses for syndrome polynomial 1 of the 2 first registers:	ELM_ERROR_LOCATION_0_i (i=1)	0x4
	ELM_ERROR_LOCATION_1_i (i=1)	0x1036
Read the errors location bit addresses for syndrome polynomial 2 of the first registers:	ELM_ERROR_LOCATION_0_i (i=1)	0x3E8
Clear the ELM_IRQSTATUS register.	ELM_IRQSTATUS	0x1FF

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.4.5 ELM Registers

Table 6-163. ELM Instance Summary

Module Name	Base Address	Size
ELM	0x4808 0000	64KB

Table 6-164. ELM Registers Mapping Summary

Address Offset	Acronym	Description	Section
000	ELM_REVISION	ELM Revision Register	Section 6.4.5.1
010h	ELM_SYSCONFIG	ELM System Configuration Register	Section 6.4.5.2
014h	ELM_SYSSTATUS	ELM System Status Register	Section 6.4.5.3
018h	ELM_IRQSTATUS	ELM Interrupt Status Register	Section 6.4.5.4
01Ch	ELM_IRQENABLE	ELM Interrupt Enable Register	Section 6.4.5.5
020h	ELM_LOCATION_CONFIG	ELM Location Configuration Register	Section 6.4.5.6
080h	ELM_PAGE_CTRL	ELM Page Definition Register	Section 6.4.5.7
400h + (40h × i)	ELM_SYNDROME_FRAGMENT_0_i	ELM_SYNDROME_FRAGMENT_0_i Register	Section 6.4.5.8
404h+ (40h × i)	ELM_SYNDROME_FRAGMENT_1_i	ELM_SYNDROME_FRAGMENT_1_i Register	Section 6.4.5.9
408h + (40h × i)	ELM_SYNDROME_FRAGMENT_2_i	ELM_SYNDROME_FRAGMENT_2_i Register	Section 6.4.5.10
40Ch + (40h × i)	ELM_SYNDROME_FRAGMENT_3_i	ELM_SYNDROME_FRAGMENT_3_i Register	Section 6.4.5.11
410h + (40h × i)	ELM_SYNDROME_FRAGMENT_4_i	ELM_SYNDROME_FRAGMENT_4_i Register	Section 6.4.5.12
414h + (40h × i)	ELM_SYNDROME_FRAGMENT_5_i	ELM_SYNDROME_FRAGMENT_5_i Register	Section 6.4.5.13
418h + (40h × i)	ELM_SYNDROME_FRAGMENT_6_1	ELM_SYNDROME_FRAGMENT_6_1 Register	Section 6.4.5.14
800h + (100h × i)	ELM_LOCATION_STATUS_i	ELM_LOCATION_STATUS_i Register	Section 6.4.5.15
880h-8BCh + (100h × i)	ELM_ERROR_LOCATION_0-15_i	ELM_ERROR_LOCATION_0-15_i Registers	Section 6.4.5.16

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.4.5.1 ELM Revision Register (ELM_REVISION)

This register contains the IP revision code. The ELM Revision Register (ELM_REVISION) is shown in Figure 6-138 and described in Table 6-165.

Figure 6-138. ELM Revision Register (ELM_REVISION)

31	0
REVISION	
R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 6-165. ELM Revision Register (ELM_REVISION) Field Descriptions

Bit	Field	Value	Description
31-0	REVISION	0-FFFF FFFFh	IP Revision

6.4.5.2 ELM System Configuration Register (ELM_SYSCONFIG)

This register allows controlling various parameters of the OCP interface. The ELM System Configuration Register (ELM_SYSCONFIG) is shown in Figure 6-139 and described in Table 6-166.

Figure 6-139. ELM System Configuration Register (ELM_SYSCONFIG)

31	16
Reserved	
R-0	

15 9	8	7 5	4 3	2	1	0
Reserved	CLOCKACTIVITYOCP	Reserved	SIDLEMODE	Reserved	SOFTRESET	AUTOGATING
R-0	R/W-0	R-0	R/W-2	R-0	R/W-0	R/W-1

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 6-166. ELM System Configuration Register (ELM_SYSCONFIG) Field Descriptions

Bit	Field	Value	Description
31-9	Reserved	0	Reserved
8	CLOCKACTIVITY OCP		OCP Clock activity when module is in IDLE mode (during wake up mode period).
		0	OCP Clock can be switch-off
		1	OCP Clock is maintained during wake up period
7-5	Reserved	0	Reserved
4-3	SIDLEMODE		Slave interface power management (IDLE req/ack control).
		0	FORCE Idle. IDLE request is acknowledged unconditionally and immediately. (Default *Dumb* mode for safety)
		1h	NO idle. IDLE request is never acknowledged.
		2h	SMART Idle. The acknowledgment to an IDLE request is given based on the internal activity.
		3h	Reserved - do not use
2	Reserved	0	Reserved
1	SOFTRESET		Module software reset. This bit is automatically reset by hardware (During reads, it always returns 0.). It has same effect as the OCP hardware reset.
		0	Normal mode.
		1	Start soft reset sequence.
0	AUTOGATING		Internal OCP clock gating strategy. (No module visible impact other than saving power.)
		0	OCP clock is free-running.
		1	Automatic internal OCP clock gating strategy is applied based on the OCP interface activity.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.4.5.3 ELM System Status Register (ELM_SYSSTATUS)

The ELM System Status Register (ELM_SYSSTATUS) is shown in Figure 6-140 and described in Table 6-167.

Figure 6-140. ELM System Status Register (ELM_SYSSTATUS)

31	1	0
Reserved		RESETDONE
R-0		R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 6-167. ELM System Status Register (ELM_SYSSTATUS) Field Descriptions

Bit	Field	Value	Description
31-1	Reserved	0	Reserved
0			Internal reset monitoring (OCP domain). Undefined since: From hardware perspective, the reset state is 0. From software user perspective, when the accessible module is 1.
		0	Reset is on-going
		1	Reset is done (completed)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.4.5.4 ELM Interrupt Status Register (ELM_IRQSTATUS)

This register doubles as a status register for the error-location processes. The ELM Interrupt Status Register (ELM_IRQSTATUS) is shown in Figure 6-141 and described in Table 6-168.

Figure 6-141. ELM Interrupt Status Register (ELM_IRQSTATUS)

31							16
Reserved							
R-0							

15						9	8
Reserved							PAGE_VALID
R-0							R/W-0

7	6	5	4	3	2	1	0
LOC_VALID_7	LOC_VALID_6	LOC_VALID_5	LOC_VALID_4	LOC_VALID_3	LOC_VALID_2	LOC_VALID_1	LOC_VALID_0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 6-168. ELM Interrupt Status Register (ELM_IRQSTATUS) Field Descriptions

Bit	Field	Value	Description
31-9	Reserved	0	Reserved
8	PAGE_VALID		Error-location status for a full page, based on the mask definition.
		0	Read: Error locations invalid for all polynomials enabled in the ECC_INTERRUPT_MASK register.
		1	Read: All error locations valid.
		0	Write: No effect.
		1	Write: Clear interrupt.
7	LOC_VALID_7		Error-location status for syndrome polynomial 7.
		0	Read: No syndrome processed or process in progress.
		1	Read: Error-location process completed.
		0	Write: No effect.
		1	Write: Clear interrupt.
6	LOC_VALID_6		Error-location status for syndrome polynomial 6.
		0	Read: No syndrome processed or process in progress.
		1	Read: Error-location process completed.
		0	Write: No effect.
		1	Write: Clear interrupt.
5	LOC_VALID_5		Error-location status for syndrome polynomial 5.
		0	Read: No syndrome processed or process in progress.
		1	Read: Error-location process completed.
		0	Write: No effect.
		1	Write: Clear interrupt.
4	LOC_VALID_4		Error-location status for syndrome polynomial 4.
		0	Read: No syndrome processed or process in progress.
		1	Read: Error-location process completed.
		0	Write: No effect.
		1	Write: Clear interrupt.
3	LOC_VALID_3		Error-location status for syndrome polynomial 3.
		0	Read: No syndrome processed or process in progress.
		1	Read: Error-location process completed.
		0	Write: No effect.
		1	Write: Clear interrupt.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 6-168. ELM Interrupt Status Register (ELM_IRQSTATUS) Field Descriptions (continued)

Bit	Field	Value	Description
2	LOC_VALID_2		Error-location status for syndrome polynomial 2.
		0	Read: No syndrome processed or process in progress.
		1	Read: Error-location process completed.
		0	Write: No effect.
		1	Write: Clear interrupt.
1	LOC_VALID_1		Error-location status for syndrome polynomial 1.
		0	Read: No syndrome processed or process in progress.
		1	Read: Error-location process completed.
		0	Write: No effect.
		1	Write: Clear interrupt.
0	LOC_VALID_0		Error-location status for syndrome polynomial 0.
		0	Read: No syndrome processed or process in progress.
		1	Read: Error-location process completed.
		0	Write: No effect.
		1	Write: Clear interrupt.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.4.5.5 ELM Interrupt Enable Register (ELM_IRQENABLE)

The ELM Interrupt Enable Register (ELM_IRQENABLE) is shown in Figure 6-142 and described in Table 6-169.

Figure 6-142. ELM Interrupt Enable Register (ELM_IRQENABLE)

31							16
Reserved							
R-0							

15						9	8
Reserved							PAGE_MASK
R-0							R/W-0

7	6	5	4	3	2	1	0
LOCATION_MASK_7	LOCATION_MASK_6	LOCATION_MASK_5	LOCATION_MASK_4	LOCATION_MASK_3	LOCATION_MASK_2	LOCATION_MASK_1	LOCATION_MASK_0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 6-169. ELM Interrupt Enable Register (ELM_IRQENABLE) Field Descriptions

Bit	Field	Value	Description
31-9	Reserved	0	Reserved
8	PAGE_MASK		Page interrupt mask bit
		0	Disable interrupt.
		1	Enable interrupt.
7	LOCATION_MASK_7		Error-location interrupt mask bit for syndrome polynomial 7.
		0	Disable interrupt.
		1	Enable interrupt.
6	LOCATION_MASK_6		Error-location interrupt mask bit for syndrome polynomial 6.
		0	Disable interrupt.
		1	Enable interrupt.
5	LOCATION_MASK_5		Error-location interrupt mask bit for syndrome polynomial 5.
		0	Disable interrupt.
		1	Enable interrupt.
4	LOCATION_MASK_4		Error-location interrupt mask bit for syndrome polynomial 4.
		0	Disable interrupt.
		1	Enable interrupt.
3	LOCATION_MASK_3		Error-location interrupt mask bit for syndrome polynomial 3.
		0	Disable interrupt.
		1	Enable interrupt.
2	LOCATION_MASK_2		Error-location interrupt mask bit for syndrome polynomial 2.
		0	Disable interrupt.
		1	Enable interrupt.
1	LOCATION_MASK_1		Error-location interrupt mask bit for syndrome polynomial 1.
		0	Disable interrupt.
		1	Enable interrupt.
0	LOCATION_MASK_0		Error-location interrupt mask bit for syndrome polynomial 0.
		0	Disable interrupt.
		1	Enable interrupt.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.4.5.6 ELM Location Configuration Register (ELM_LOCATION_CONFIG)

The ELM Location Configuration Register (ELM_LOCATION_CONFIG) is shown in Figure 6-143 and described in Table 6-170.

Figure 6-143. ELM Location Configuration Register (ELM_LOCATION_CONFIG)

31–27	26–16	15–2	1–0
Reserved	ECC_SIZE	Reserved	ECC_BCH_LEVEL
R-0	R/W-0	R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 6-170. ELM Location Configuration Register (ELM_LOCATION_CONFIG) Field Descriptions

Bit	Field	Value	Description
31-27	Reserved	0	Reserved
26-16	ECC_SIZE	0-7FFh	Maximum size of the buffers for which the error-location engine is used, in number of nibbles (4-bits entities)
15-2	Reserved	0	Reserved
1-0	ECC_BCH_LEVEL		Error correction level.
		0	4 bits.
		1h	8 bits.
		2h	16 bits.
		3h	Reserved.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.4.5.7 ELM Page Definition Register (ELM_PAGE_CTRL)

The ELM Page Definition Register (ELM_PAGE_CTRL) is shown in Figure 6-144 and described in Table 6-171.

Figure 6-144. ELM Page Definition Register (ELM_PAGE_CTRL)

31							8
Reserved							
R-0							
7	6	5	4	3	2	1	0
SECTOR_7	SECTOR_6	SECTOR_5	SECTOR_4	SECTOR_3	SECTOR_2	SECTOR_1	SECTOR_0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 6-171. ELM Page Definition Register (ELM_PAGE_CTRL) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved.
7	SECTOR_7	0-1	Set to 1 if syndrome polynomial 7 is part of the page in page mode. Must be 0 in continuous mode.
6	SECTOR_6	0-1	Set to 1 if syndrome polynomial 6 is part of the page in page mode. Must be 0 in continuous mode.
5	SECTOR_5	0-1	Set to 1 if syndrome polynomial 5 is part of the page in page mode. Must be 0 in continuous mode.
4	SECTOR_4	0-1	Set to 1 if syndrome polynomial 4 is part of the page in page mode. Must be 0 in continuous mode.
3	SECTOR_3	0-1	Set to 1 if syndrome polynomial 3 is part of the page in page mode. Must be 0 in continuous mode.
2	SECTOR_2	0-1	Set to 1 if syndrome polynomial 2 is part of the page in page mode. Must be 0 in continuous mode.
1	SECTOR_1	0-1	Set to 1 if syndrome polynomial 1 is part of the page in page mode. Must be 0 in continuous mode.
0	SECTOR_0	0-1	Set to 1 if syndrome polynomial 0 is part of the page in page mode. Must be 0 in continuous mode.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.4.5.8 ELM_SYNDROME_FRAGMENT_0_i Register

The ELM_SYNDROME_FRAGMENT_0_i Register is shown in Figure 6-145 and described in Table 6-172.

Figure 6-145. ELM_SYNDROME_FRAGMENT_0_i Register

31	0
SYNDROME_0	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 6-172. ELM_SYNDROME_FRAGMENT_0_i Register Field Descriptions

Bit	Field	Value	Description
31-0	SYNDROME_0	0-FFFF FFFFh	Syndrome bits 0 to 31.

6.4.5.9 ELM_SYNDROME_FRAGMENT_1_i Register

The ELM_SYNDROME_FRAGMENT_1_i Register is shown in Figure 6-146 and described in Table 6-173.

Figure 6-146. ELM_SYNDROME_FRAGMENT_1_i Register

31	0
SYNDROME_1	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 6-173. ELM_SYNDROME_FRAGMENT_1_i Register Field Descriptions

Bit	Field	Value	Description
31-0	SYNDROME_1	0-FFFF FFFFh	Syndrome bits 32 to 63.

6.4.5.10 ELM_SYNDROME_FRAGMENT_2_i Register

The ELM_SYNDROME_FRAGMENT_2_i Register is shown in Figure 6-147 and described in Table 6-174.

Figure 6-147. ELM_SYNDROME_FRAGMENT_2_i Register

31	0
SYNDROME_2	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 6-174. ELM_SYNDROME_FRAGMENT_2_i Register Field Descriptions

Bit	Field	Value	Description
31-0	SYNDROME_2	0-FFFF FFFFh	Syndrome bits 64 to 95.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.4.5.11 ELM_SYNDROME_FRAGMENT_3_i Register

The ELM_SYNDROME_FRAGMENT_3_i Register is shown in Figure 6-148 and described in Table 6-175.

Figure 6-148. ELM_SYNDROME_FRAGMENT_3_i Register

31	0
SYNDROME_3	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 6-175. ELM_SYNDROME_FRAGMENT_3_i Register Field Descriptions

Bit	Field	Value	Description
31-0	SYNDROME_3	0-FFFF FFFFh	Syndrome bits 96 to 127.

6.4.5.12 ELM_SYNDROME_FRAGMENT_4_i Register

The ELM_SYNDROME_FRAGMENT_4_i Register is shown in Figure 6-149 and described in Table 6-176.

Figure 6-149. ELM_SYNDROME_FRAGMENT_4_i Register

31	0
SYNDROME_4	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 6-176. ELM_SYNDROME_FRAGMENT_4_i Register Field Descriptions

Bit	Field	Value	Description
31-0	SYNDROME_4	0-FFFF FFFFh	Syndrome bits 128 to 159.

6.4.5.13 ELM_SYNDROME_FRAGMENT_5_i Register

The ELM_SYNDROME_FRAGMENT_5_i Register is shown in Figure 6-150 and described in Table 6-177.

Figure 6-150. ELM_SYNDROME_FRAGMENT_5_i Register

31	0
SYNDROME_5	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 6-177. ELM_SYNDROME_FRAGMENT_5_i Register Field Descriptions

Bit	Field	Value	Description
31-0	SYNDROME_5	0-FFFF FFFFh	Syndrome bits 160 to 191.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.4.5.14 ELM_SYNDROME_FRAGMENT_6_i Register

The ELM_SYNDROME_FRAGMENT_6_i Register is shown in Figure 6-151 and described in Table 6-178.

Figure 6-151. ELM_SYNDROME_FRAGMENT_6_i Register

31–17	16	15–0
Reserved	SYNDROME_VALID	SYNDROME_6
R-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 6-178. ELM_SYNDROME_FRAGMENT_6_i Register Field Descriptions

Bit	Field	Value	Description
31-17	Reserved	0	Reserved
16	SYNDROME_VALID		Syndrome valid bit.
		0	This syndrome polynomial should not be processed.
		1	This syndrome polynomial must be processed.
15-0	SYNDROME_6	0-FFFFh	Syndrome bits 192 to 207.

6.4.5.15 ELM_LOCATION_STATUS_i Register

The ELM_LOCATION_STATUS_i Register is shown in Figure 6-152 and described in Table 6-179.

Figure 6-152. ELM_LOCATION_STATUS_i Register

31–9	8	7–5	4–0
Reserved	ECC_CORRECTABL	Reserved	ECC_NB_ERRORS
R-0	R-0	R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 6-179. ELM_LOCATION_STATUS_i Register Field Descriptions

Bit	Field	Value	Description
31-9	Reserved	0	Reserved
8	ECC_CORRECTABL		Error-location process exit status.
		0	ECC error-location process failed. Number of errors and error locations are invalid.
		1	All errors were successfully located. Number of errors and error locations are valid.
7-5	Reserved	0	Reserved
4-0	ECC_NB_ERRORS	0-1Fh	Number of errors detected and located.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

6.4.5.16 ELM_ERROR_LOCATION_0-15_i Registers

The ELM_ERROR_LOCATION_0-15_i Registers is shown in Figure 6-153 and described in Table 6-180.

Figure 6-153. ELM_ERROR_LOCATION_0-15_i Registers

31 – 13	12 – 0
Reserved	ECC_ERROR_LOCATION
R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 6-180. ELM_ERROR_LOCATION_0-15_i Registers Field Descriptions

Bit	Field	Value	Description
31-13	Reserved	0	Reserved
12-0	ECC_ERROR_LOCATION	0-1FFFh	Error-location bit address.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Chapter 7

SPRUH73–July 2011

Power and Clock Management (PRCM)

This chapter describes the PRCM of the device.

Topic **Page**

7.1 Power Management **1084**

7.2 Clock Management **1113**

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.1 Power Management

7.1.1 Introduction

7.1.2 Integration

7.1.3 Functional Description

7.1.4 Use Cases

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.1.5 Registers

7.1.5.1 PRM_CEFUSE Registers

Table 7-1 lists the memory-mapped registers for the PRM_CEFUSE. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 7-1 should be considered as reserved locations and the register contents should not be modified.

Table 7-1. PRM_CEFUSE REGISTERS

Offset	Acronym	Register Description	Section
0h	PM_CEFUSE_PWRSTCTRL	This register controls the CEFUSE power state to reach upon a domain sleep transition	Section 7.1.5.1.1
4h	PM_CEFUSE_PWRSTST	This register provides a status on the current CEFUSE power domain state. [warm reset insensitive]	Section 7.1.5.1.2

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.1.5.1.1 PM_CEFUSE_PWRSTCTRL Register (offset = 0h)

This register controls the CEFUSE power state to reach upon a domain sleep transition

PM_CEFUSE_PWRSTCTRL is shown in Figure 7-1 and described in Table 7-2.

Figure 7-1. PM_CEFUSE_PWRSTCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-2. PM_CEFUSE_PWRSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	Reserved	R	0h	
4	LowPowerStateChange	R/W	0h	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0 - DIS: Do not request a low power state change. 1 - EN: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.
3-2	Reserved	R	0h	
1-0	PowerState	R/W	0h	Power state control 0 - OFF: OFF state 1 - Reserved: Reserved 2 - INACT: INACTIVE state 3 - ON: ON State

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.1.5.1.2 PM_CEFUSE_PWRSTST Register (offset = 4h)

This register provides a status on the current CEFUSE power domain state. [warm reset insensitive]

PM_CEFUSE_PWRSTST is shown in Figure 7-2 and described in Table 7-3.

Figure 7-2. PM_CEFUSE_PWRSTST Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-3. PM_CEFUSE_PWRSTST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	R	0h	
25-24	LastPowerStateEntered	R/W	0h	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0 - OFF: Power domain was previously OFF 1 - ON: Power domain was previously ON-ACTIVE
23-21	Reserved	R	0h	
20	InTransition	R	0h	Domain transition status 0 - No: No on-going transition on power domain 1 - Ongoing: Power domain transition is in progress.
19-3	Reserved	R	0h	
2	LogicStateSt	R	1h	Logic state status 0 - OFF: Logic in domain is OFF 1 - ON: Logic in domain is ON
1-0	PowerStateSt	R	3h	Current power state status 0 - OFF: Power domain is OFF 1 - RET: Power domain is in RETENTION 2 - INACTIVE: Power domain is ON-INACTIVE 3 - ON: Power domain is ON-ACTIVE

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.1.5.2 PRM_DEVICE Registers

Table 7-4 lists the memory-mapped registers for the PRM_DEVICE. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 7-4 should be considered as reserved locations and the register contents should not be modified.

Table 7-4. PRM_DEVICE REGISTERS

Offset	Acronym	Register Description	Section
0h	PRM_RSTCTRL	Global software cold and warm reset control. This register is auto-cleared. Only write 1 is possible. A read returns 0 only.	Section 7.1.5.2.1
Ch	PRM_SRAM_COUNT	Common setup for SRAM LDO transition counters. Applies to all voltage domains. [warm reset insensitive]	Section 7.1.5.2.2
10h	PRM_LDO_SRAM_CORE_SETUP	Setup of the SRAM LDO for CORE voltage domain. [warm reset insensitive]	Section 7.1.5.2.3
14h	PRM_LDO_SRAM_CORE_CTRL	Control and status of the SRAM LDO for CORE voltage domain. [warm reset insensitive]	Section 7.1.5.2.4
18h	PRM_LDO_SRAM_MPU_SETUP	Setup of the SRAM LDO for MPU voltage domain. [warm reset insensitive]	Section 7.1.5.2.5
1Ch	PRM_LDO_SRAM_MPU_CTRL	Control and status of the SRAM LDO for MPU voltage domain. [warm reset insensitive]	Section 7.1.5.2.6
4h	PRM_RSTTIME	Reset duration control. [warm reset insensitive]	Section 7.1.5.2.7
8h	PRM_RSTST	This register logs the global reset sources. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]	Section 7.1.5.2.8

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.1.5.2.1 PRM_RSTCTRL Register (offset = 0h)

Global software cold and warm reset control. This register is auto-cleared. Only write 1 is possible. A read returns 0 only.

PRM_RSTCTRL is shown in Figure 7-3 and described in Table 7-5.

Figure 7-3. PRM_RSTCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-5. PRM_RSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	Reserved	R	0h	
1	RST_GLOBAL_COLD_SW	R/W	0h	Global COLD software reset control. This bit is reset only upon a global cold source of reset. 0 - 0x0: Global COLD software reset is cleared. 1 - 0x1: Asserts a global COLD software reset. The software must ensure the SDRAM is properly put in sef-refresh mode before applying this reset.
0	RST_GLOBAL_WARM_SW	R/W	0h	Global WARM software reset control. This bit is reset upon any global source of reset (warm and cold). 0 - 0x0: Global warm software reset is cleared. 1 - 0x1: Asserts a global warm software reset.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.1.5.2.2 PRM_SRAM_COUNT Register (offset = Ch)

Common setup for SRAM LDO transition counters. Applies to all voltage domains. [warm reset insensitive]

PRM_SRAM_COUNT is shown in Figure 7-4 and described in Table 7-6.

Figure 7-4. PRM_SRAM_COUNT Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-6. PRM_SRAM_COUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	StartUp_Count	R/W	78h	Determines the start-up duration of SRAM and ABB LDO. The duration is computed as 16 x NbCycles of system clock cycles. Target is 50us.
23-16	SLPCNT_VALUE	R/W	0h	Delay between retention/off assertion of last SRAM bank and SRAMALLRET signal to LDO is driven high. Counting on system clock. Target is 2us.
15-8	VSETUPCNT_VALUE	R/W	0h	SRAM LDO rampup time from retention to active mode. The duration is computed as 8 x NbCycles of system clock cycles. Target is 30us.
7-6	Reserved	R	0h	
5-0	PCHARGECNT_VALUE	R/W	17h	Delay between de-assertion of standby_rta_ret_on and standby_rta_ret_good. Counting on system clock. Target is 600ns.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.1.5.2.3 PRM_LDO_SRAM_CORE_SETUP Register (offset = 10h)

Setup of the SRAM LDO for CORE voltage domain. [warm reset insensitive]

PRM_LDO_SRAM_CORE_SETUP is shown in Figure 7-5 and described in Table 7-7.

Figure 7-5. PRM_LDO_SRAM_CORE_SETUP Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-7. PRM_LDO_SRAM_CORE_SETUP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R	0h	
8	AIPOFF	R/W	0h	Override on AIPOFF input of SRAM LDO. 0 - No_Override: AIPOFF signal is not overriden 1 - Override: AIPOFF signal is overriden to '1'. Corresponding SRAM LDO is disabled and in HZ mode.
7	ENFUNC5	R/W	0h	ENFUNC5 input of SRAM LDO. 0 - One_step: Active to retention is a one step transfer 1 - Two_step: Active to retention is a two steps transfer
6	ENFUNC4	R/W	0h	ENFUNC4 input of SRAM LDO. 0 - Ext_clock: One external clock is supplied 1 - No_ext_clock: No external clock is supplied
5	ENFUNC3_EXPORT	R/W	0h	ENFUNC3 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0 - Sub_regul_disabled: Sub regulation is disabled 1 - Sub_regul_enabled: Sub regulation is enabled
4	ENFUNC2_EXPORT	R/W	0h	ENFUNC2 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0 - Ext_cap: External cap is used 1 - No_ext_cap: External cap is not used
3	ENFUNC1_EXPORT	R/W	0h	ENFUNC1 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0 - Short_prot_disabled: Short circuit protection is disabled 1 - Short_prot_enabled: Short circuit protection is enabled
2	ABBOFF_SLEEP_EXPORT	R/W	0h	Determines whether SRAMNWA is supplied by VDDS or VDDAR during deep-sleep. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0 - SRAMNW_SLP_VDDS: SRAMNWA supplied with VDDS 1 - SRAMNW_SLP_VDDAR: SRAMNWA supplied with VDDAR
1	ABBOFF_ACT_EXPORT	R/W	0h	Determines whether SRAMNWA is supplied by VDDS or VDDAR during active mode. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0 - SRAMNW_ACT_VDDS: SRAMNWA supplied with VDDS 1 - SRAMNW_ACT_VDDAR: SRAMNWA supplied with VDDAR
0	DISABLE_RTA_EXPORT	R/W	0h	Control for HD memory RTA feature. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0 - RTA_ENABLED: HD memory RTA feature is enabled 1 - RTA_DISABLED: HD memory RTA feature is disabled

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.1.5.2.4 PRM_LDO_SRAM_CORE_CTRL Register (offset = 14h)

Control and status of the SRAM LDO for CORE voltage domain. [warm reset insensitive]

PRM_LDO_SRAM_CORE_CTRL is shown in Figure 7-6 and described in Table 7-8.

Figure 7-6. PRM_LDO_SRAM_CORE_CTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-8. PRM_LDO_SRAM_CORE_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	Reserved	R	0h	
9	SRAM_IN_TRANSITION	R	0h	Status indicating SRAM LDO state machine state. 0 - IDLE: SRAM LDO state machine is stable 1 - IN_TRANSITION: SRAM LDO state machine is in transition state
8	SRAMLDO_STATUS	R	0h	SRAMLDO status 0 - ACTIVE: SRAMLDO is in ACTIVE mode. 1 - RETENTION: SRAMLDO is on RETENTION mode.
7-1	Reserved	R	0h	
0	RETMODE_ENABLE	R/W	0h	Control if the SRAM LDO retention mode is used or not. 0 - Disabled: SRAM LDO is not allowed to go to RET mode 1 - Enabled: SRAM LDO go to RET mode when all memory of voltage domain are OFF or RET

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.1.5.2.5 PRM_LDO_SRAM_MPU_SETUP Register (offset = 18h)

Setup of the SRAM LDO for MPU voltage domain. [warm reset insensitive]

PRM_LDO_SRAM_MPU_SETUP is shown in Figure 7-7 and described in Table 7-9.

Figure 7-7. PRM_LDO_SRAM_MPU_SETUP Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-9. PRM_LDO_SRAM_MPU_SETUP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R	0h	
8	AIPOFF	R/W	0h	Override on AIPOFF input of SRAM LDO. 0 - No_Override: AIPOFF signal is not overriden 1 - Override: AIPOFF signal is overriden to '1'. Corresponding SRAM LDO is disabled and in HZ mode.
7	ENFUNC5	R/W	0h	ENFUNC5 input of SRAM LDO. 0 - One_step: Active to retention is a one step transfer 1 - Two_step: Active to retention is a two steps transfer
6	ENFUNC4	R/W	0h	ENFUNC4 input of SRAM LDO. 0 - Ext_clock: One external clock is supplied 1 - No_ext_clock: No external clock is supplied
5	ENFUNC3_EXPORT	R/W	0h	ENFUNC3 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0 - Sub_regul_disabled: Sub regulation is disabled 1 - Sub_regul_enabled: Sub regulation is enabled
4	ENFUNC2_EXPORT	R/W	0h	ENFUNC2 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0 - Ext_cap: External cap is used 1 - No_ext_cap: External cap is not used
3	ENFUNC1_EXPORT	R/W	0h	ENFUNC1 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0 - Short_prot_disabled: Short circuit protection is disabled 1 - Short_prot_enabled: Short circuit protection is enabled
2	ABBOFF_SLEEP_EXPORT	R/W	0h	Determines whether SRAMNWA is supplied by VDDS or VDDAR during deep-sleep. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0 - SRAMNW_SLP_VDDS: SRAMNWA supplied with VDDS 1 - SRAMNW_SLP_VDDAR: SRAMNWA supplied with VDDAR
1	ABBOFF_ACT_EXPORT	R/W	0h	Determines whether SRAMNWA is supplied by VDDS or VDDAR during active mode. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0 - SRAMNW_ACT_VDDS: SRAMNWA supplied with VDDS 1 - SRAMNW_ACT_VDDAR: SRAMNWA supplied with VDDAR
0	DISABLE_RTA_EXPORT	R/W	0h	Control for HD memory RTA feature. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0 - RTA_ENABLED: HD memory RTA feature is enabled 1 - RTA_DISABLED: HD memory RTA feature is disabled

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.1.5.2.6 PRM_LDO_SRAM_MPU_CTRL Register (offset = 1Ch)

Control and status of the SRAM LDO for MPU voltage domain. [warm reset insensitive]

PRM_LDO_SRAM_MPU_CTRL is shown in Figure 7-8 and described in Table 7-10.

Figure 7-8. PRM_LDO_SRAM_MPU_CTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-10. PRM_LDO_SRAM_MPU_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	Reserved	R	0h	
9	SRAM_IN_TRANSITION	R	0h	Status indicating SRAM LDO state machine state. 0 - IDLE: SRAM LDO state machine is stable 1 - IN_TRANSITION: SRAM LDO state machine is in transition state
8	SRAMLDO_STATUS	R	0h	SRAMLDO status 0 - ACTIVE: SRAMLDO is in ACTIVE mode. 1 - RETENTION: SRAMLDO is on RETENTION mode.
7-1	Reserved	R	0h	
0	RETMODE_ENABLE	R/W	0h	Control if the SRAM LDO retention mode is used or not. 0 - Disabled: SRAM LDO is not allowed to go to RET mode 1 - Enabled: SRAM LDO go to RET mode when all memory of voltage domain are OFF or RET

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.1.5.2.7 PRM_RSTTIME Register (offset = 4h)

Reset duration control. [warm reset insensitive]

PRM_RSTTIME is shown in Figure 7-9 and described in Table 7-11.

Figure 7-9. PRM_RSTTIME Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-11. PRM_RSTTIME Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	Reserved	R	0h	
12-8	RSTTIME2	R/W	10h	(Power domain) reset duration 2 (number of RM.SYSCLK clock cycles)
7-0	RSTTIME1	R/W	06h	(Global) reset duration 1 (number of SYS_CLK clock cycles)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.1.5.2.8 PRM_RSTST Register (offset = 8h)

This register logs the global reset sources. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]

PRM_RSTST is shown in Figure 7-10 and described in Table 7-12.

Figure 7-10. PRM_RSTST Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-12. PRM_RSTST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	Reserved	R	0h	
9	ICEPICK_RST	R/W	0h	IcePick reset event. This is a source of global warm reset initiated by the emulation. [warm reset insensitive] 0 - 0x0: No ICEPICK reset. 1 - 0x1: IcePick reset has occurred.
8-6	Reserved	R	0h	
5	EXTERNAL_WARM_RST	R/W	0h	External warm reset event [warm reset insensitive] 0 - 0x0: No global warm reset. 1 - 0x1: Global external warm reset has occurred.
4	WDT1_RST	R/W	0h	Watchdog1 timer reset event. This is a source of global WARM reset. [warm reset insensitive] 0 - 0x0: No watchdog reset. 1 - 0x1: watchdog reset has occurred.
3	WDT0_RST	R/W	0h	Watchdog0 timer reset event. This is a source of global WARM reset. [warm reset insensitive] 0 - 0x0: No watchdog reset. 1 - 0x1: watchdog reset has occurred.
2	MPU_SECURITY_VIOL_RST	R/W	0h	Security violation reset event triggered by Security State Machine inside MPUSS. This is a source of global WARM reset. [warm reset insensitive] 0 - 0x0: No security violation reset. 1 - 0x1: Security violation reset has occurred.
1	GLOBAL_WARM_SW_RST	R/W	0h	Global warm software reset event [warm reset insensitive] 0 - 0x0: No global warm SW reset 1 - 0x1: Global warm SW reset has occurred.
0	GLOBAL_COLD_RST	R/W	1h	Power-on (cold) reset event [warm reset insensitive] 0 - 0x0: No power-on reset. 1 - 0x1: Power-on reset has occurred.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.1.5.3 PRM_MPU Registers

Table 7-13 lists the memory-mapped registers for the PRM_MPU. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 7-13 should be considered as reserved locations and the register contents should not be modified.

Table 7-13. PRM_MPU REGISTERS

Offset	Acronym	Register Description	Section
0h	PM_MPU_PWRSTCTRL	This register controls the MPU power state to reach upon mpu domain sleep transition	Section 7.1.5.3.1
4h	PM_MPU_PWRSTST	This register provides a status on the current MPU power domain state0. [warm reset insensitive]	Section 7.1.5.3.2
8h	RM_MPU_RSTST	This register logs the different reset sources of the ALWON domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]	Section 7.1.5.3.3

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.1.5.3.1 PM_MPU_PWRSTCTRL Register (offset = 0h)

This register controls the MPU power state to reach upon mpu domain sleep transition

PM_MPU_PWRSTCTRL is shown in Figure 7-11 and described in Table 7-14.

Figure 7-11. PM_MPU_PWRSTCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-14. PM_MPU_PWRSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	R	0h	
25	Reserved	R	0h	
24	mpu_ram_RETState	R/W	1h	Default power domain memory(ram) retention state when power domain is in retention
23	mpu_l2_RETState	R/W	1h	Default power domain memory(L2) retention state when power domain is in retention
22	mpu_l1_RETState	R/W	1h	Default power domain memory(L1) retention state when power domain is in retention
21-20	MPU_L2_ONState	R	3h	Default power domain memory state when domain is ON. 3 - Mem_on: Memory bank is on when the domain is ON.
19-18	MPU_L1_ONState	R	3h	Default power domain memory state when domain is ON. 3 - Mem_on: Memory bank is on when the domain is ON.
17-16	MPU_RAM_ONState	R/W	3h	Default power domain memory state when domain is ON. 0 - Mem_off 2 - Reserved 3 - Mem_on: Memory bank is on when the domain is ON.
15-5	Reserved	R	0h	
4	LowPowerStateChange	R/W	0h	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0 - DIS: Do not request a low power state change. 1 - EN: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.
3	Reserved	R	0h	
2	LogicRETState	R/W	1h	Logic state when power domain is RETENTION 0 - logic_off: Only retention registers are retained and remaing logic is off when the domain is in RETENTION state. 1 - logic_ret: Whole logic is retained when domain is in RETENTION state.
1-0	PowerState	R/W	3h	Power state control 0 - OFF: OFF State 1 - RET 2 - Reserved 3 - ON: ON State

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.1.5.3.2 PM_MPU_PWRSTST Register (offset = 4h)

This register provides a status on the current MPU power domain state0. [warm reset insensitive]

PM_MPU_PWRSTST is shown in Figure 7-12 and described in Table 7-15.

Figure 7-12. PM_MPU_PWRSTST Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-15. PM_MPU_PWRSTST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	Reserved	R	0h	
20	InTransition	R	0h	Domain transition status 0 - No: No on-going transition on power domain 1 - Ongoing: Power domain transition is in progress.
19-10	Reserved	R	0h	
9-8	MPU_L2_StateSt	R	1h	MPU L2 memory state status 0 - Mem_off: Memory is OFF 2 - Reserved: Reserved 3 - Mem_on: Memory is ON
7-6	MPU_L1_StateSt	R	1h	MPU L1 memory state status 0 - Mem_off: Memory is OFF 2 - Reserved: Reserved 3 - Mem_on: Memory is ON
5-4	MPU_RAM_StateSt	R	1h	MPU_RAM memory state status 0 - Mem_off: Memory is OFF 2 - Reserved: Reserved 3 - Mem_on: Memory is ON
3	Reserved	R	0h	
2	LogicStateSt	R	1h	Logic state status 0 - OFF: Logic in domain is OFF 1 - ON: Logic in domain is ON
1-0	PowerStateSt	R	3h	Current Power State Status 0 - OFF: OFF State [warm reset insensitive] 1 - RET: RET State 3 - ON: ON State [warm reset insensitive]

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.1.5.3.3 RM_MPU_RSTST Register (offset = 8h)

This register logs the different reset sources of the ALWON domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]

RM_MPU_RSTST is shown in Figure 7-13 and described in Table 7-16.

Figure 7-13. RM_MPU_RSTST Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-16. RM_MPU_RSTST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	Reserved	R	0h	
14-8	Reserved	R	0h	
7	Reserved	R	0h	
6	ICECRUSHER_MPU_RST	R/W	0h	MPU Processor has been reset due to MPU ICECRUSHER1 reset event 0 - RESET_NO: No icecrusher reset 1 - RESET_YES: MPU Processor has been reset upon icecursher reset
5	EMULATION_MPU_RST	R/W	0h	MPU Processor has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module 0 - RESET_NO: No emulation reset 1 - RESET_YES: MPU Processor has been reset upon emulation reset
4	Reserved	R	0h	
3	Reserved	R	0h	
2	Reserved	R	0h	
1-0	Reserved	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.1.5.4 PRM_PER Registers

Table 7-17 lists the memory-mapped registers for the PRM_PER. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 7-17 should be considered as reserved locations and the register contents should not be modified.

Table 7-17. PRM_PER REGISTERS

Offset	Acronym	Register Description	Section
0h	RM_PER_RSTCTRL	This register controls the release of the PER Domain resets.	Section 7.1.5.4.1
Ch	PM_PER_PWRSTCTRL	Controls the power state of PER power domain	Section 7.1.5.4.2
4h	RM_PER_RSTST	This register logs the different reset sources of the ALWON domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]	Section 7.1.5.4.3
8h	PM_PER_PWRSTST	This register provides a status on the current PER power domain state. [warm reset insensitive]	Section 7.1.5.4.4

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.1.5.4.1 RM_PER_RSTCTRL Register (offset = 0h)

This register controls the release of the PER Domain resets.

RM_PER_RSTCTRL is shown in Figure 7-14 and described in Table 7-18.

Figure 7-14. RM_PER_RSTCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-18. RM_PER_RSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	Reserved	R	0h	
5-3	Reserved	R	0h	
2	Reserved	R	0h	
1	ICSS_LRST	R/W	1h	PER domain ICSS local reset control 0 - CLEAR: Reset is cleared for the ICSS 1 - ASSERT: Reset is asserted for the ICSS
0	PCI_LRST	R/W	1h	PER domain PCIE local reset control 0 - CLEAR: Reset is cleared for the PCIE 1 - ASSERT: Reset is asserted for the PCIE

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.1.5.4.2 PM_PER_PWRSTCTRL Register (offset = Ch)

Controls the power state of PER power domain

PM_PER_PWRSTCTRL is shown in Figure 7-15 and described in Table 7-19.

Figure 7-15. PM_PER_PWRSTCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-19. PM_PER_PWRSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	ram_mem_ONState	R/W	3h	OCMC RAM memory on state 0 - OFF: Memory is OFF 1 - RET: Memory is in retention state 2 - RESERVED 3 - ON: Memory is ON
29	PER_mem_RETState	R/W	1h	Other memories in PER Domain RET state 0 - OFF 1 - RET
28	Reserved	R	0h	
27	ram_mem_RETState	R/W	1h	OCMC RAM memory RET state 0 - OFF: Memory is in off state 1 - RET: Memory is in retention state
26-25	PER_mem_ONState	R/W	3h	Other memories in PER Domain ON state 0 - Reserved2 1 - Reserved1 2 - Reserved: Reserved 3 - ON: Memory is ON
24-8	Reserved	R	0h	
7	icss_mem_RETState	R/W	1h	ICSS memory RET state 0 - OFF: Memory is in off state 1 - RET: Memory is in retention state
6-5	icss_mem_ONState	R/W	3h	ICSS memory ON state 0 - Reserved2 1 - Reserved1 2 - Reserved: Reserved 3 - ON: Memory is ON
4	LowPowerStateChange	R/W	0h	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0 - DIS: Do not request a low power state change. 1 - EN: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.
3	LogicRETState	R/W	1h	Logic state when power domain is RETENTION 0 - logic_off: Only retention registers are retained and remaing logic is off when the domain is in RETENTION state. 1 - logic_ret: Whole logic is retained when domain is in RETENTION state.
2	Reserved	R	0h	
1-0	PowerState	R/W	3h	PER domain power state control 0 - OFF 1 - RET 2 - Reserved 3 - ON

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.1.5.4.3 RM_PER_RSTST Register (offset = 4h)

This register logs the different reset sources of the ALWON domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]

RM_PER_RSTST is shown in Figure 7-16 and described in Table 7-20.

Figure 7-16. RM_PER_RSTST Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-20. RM_PER_RSTST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	Reserved	R	0h	
14-8	Reserved	R	0h	
7-6	Reserved	R	0h	
5	PCI_LRST	R/W	1h	PCIE has been reset upon SW reset 0 - RESET_NO: No PCIE reset 1 - RESET_YES: PCIE has been reset
4	Reserved	R	0h	
3	Reserved	R	0h	
2	Reserved	R	0h	
1-0	Reserved	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.1.5.4.4 PM_PER_PWRSTST Register (offset = 8h)

This register provides a status on the current PER power domain state. [warm reset insensitive]

PM_PER_PWRSTST is shown in Figure 7-17 and described in Table 7-21.

Figure 7-17. PM_PER_PWRSTST Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-21. PM_PER_PWRSTST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-25	Reserved	R	0h	
24-23	icss_mem_statest	R	3h	ICSS memory state status 0 - Mem_off: Memory is OFF 2 - Reserved: Reserved 3 - Mem_on: Memory is ON
22-21	ram_mem_statest	R	3h	OCMC RAM memory state status 0 - Mem_off: Memory is OFF 2 - Reserved: Reserved 3 - Mem_on: Memory is ON
20	InTransition	R	0h	Domain transition status 0 - No: No on-going transition on power domain 1 - Ongoing: Power domain transition is in progress.
19	Reserved	R	0h	
18-17	PER_mem_statest	R	3h	PER domain memory state status 0 - Mem_off: Memory is OFF 2 - Reserved: Reserved 3 - Mem_on: Memory is ON
16-3	Reserved	R	0h	
2	LogicStateSt	R	1h	Logic state status 0 - OFF: Logic in domain is OFF 1 - ON: Logic in domain is ON
1-0	PowerStateSt	R	3h	Current Power State Status 0 - OFF: OFF State 1 - RET 2 - Reserved1 3 - ON: ON State

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.1.5.5 PRM_RTC Registers

Table 7-22 lists the memory-mapped registers for the PRM_RTC. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 7-22 should be considered as reserved locations and the register contents should not be modified.

Table 7-22. PRM_RTC REGISTERS

Offset	Acronym	Register Description	Section
0h	PM_RTC_PWRSTCTRL	This register controls the RTC power state to reach upon mpu domain sleep transition	Section 7.1.5.5.1
4h	PM_RTC_PWRSTST	This register provides a status on the current RTC power domain state0. [warm reset insensitive]	Section 7.1.5.5.2

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.1.5.5.1 PM_RTC_PWRSTCTRL Register (offset = 0h)

This register controls the RTC power state to reach upon mpu domain sleep transition

PM_RTC_PWRSTCTRL is shown in Figure 7-18 and described in Table 7-23.

Figure 7-18. PM_RTC_PWRSTCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-23. PM_RTC_PWRSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	R	0h	
25-16	Reserved	R	0h	
15-5	Reserved	R	0h	
4	LowPowerStateChange	R/W	0h	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0 - DIS: Do not request a low power state change. 1 - EN: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.
3	Reserved	R	0h	
2	LogicRETState	R/W	1h	Logic state when power domain is RETENTION 0 - logic_off: Only retention registers are retained and remaing logic is off when the domain is in RETENTION state. 1 - logic_ret: Whole logic is retained when domain is in RETENTION state.
1-0	Reserved	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.1.5.5.2 PM_RTC_PWRSTST Register (offset = 4h)

This register provides a status on the current RTC power domain state0. [warm reset insensitive]

PM_RTC_PWRSTST is shown in Figure 7-19 and described in Table 7-24.

Figure 7-19. PM_RTC_PWRSTST Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-24. PM_RTC_PWRSTST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-21	Reserved	R	0h	
20	InTransition	R	0h	Domain transition status 0 - No: No on-going transition on power domain 1 - Ongoing: Power domain transition is in progress.
19-10	Reserved	R	0h	
9-4	Reserved	R	0h	
3	Reserved	R	0h	
2	LogicStateSt	R	1h	Logic state status 0 - OFF: Logic in domain is OFF 1 - ON: Logic in domain is ON
1-0	Reserved	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.1.5.6 PRM_WKUP Registers

Table 7-25 lists the memory-mapped registers for the PRM_WKUP. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 7-25 should be considered as reserved locations and the register contents should not be modified.

Table 7-25. PRM_WKUP REGISTERS

Offset	Acronym	Register Description	Section
0h	RM_WKUP_RSTCTRL	This register controls the release of the ALWAYS ON Domain resets.	Section 7.1.5.6.1
Ch	RM_WKUP_RSTST	This register logs the different reset sources of the ALWON domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]	Section 7.1.5.6.2
4h	PM_WKUP_PWRSTCTRL	Controls power state of WKUP power domain	Section 7.1.5.6.3
8h	PM_WKUP_PWRSTST	This register provides a status on the current WKUP power domain state. [warm reset insensitive]	Section 7.1.5.6.4

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.1.5.6.1 RM_WKUP_RSTCTRL Register (offset = 0h)

This register controls the release of the ALWAYS ON Domain resets.

RM_WKUP_RSTCTRL is shown in Figure 7-20 and described in Table 7-26.

Figure 7-20. RM_WKUP_RSTCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-26. RM_WKUP_RSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	Reserved	R	0h	
5-4	Reserved	R	0h	
3	WKUP_M3_LRST	R/W	1h	Assert Reset to WKUP_M3 0 - CLEAR: Reset is cleared for the M3 1 - ASSERT: Reset is asserted for the M3 by A8
2-0	Reserved	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.1.5.6.2 RM_WKUP_RSTST Register (offset = Ch)

This register logs the different reset sources of the ALWON domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]

RM_WKUP_RSTST is shown in Figure 7-21 and described in Table 7-27.

Figure 7-21. RM_WKUP_RSTST Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-27. RM_WKUP_RSTST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	Reserved	R	0h	
7	ICECRUSHER_M3_RST	R/W	0h	M3 Processor has been reset due to M3 ICECRUSHER1 reset event 0 - RESET_NO: No reset 1 - RESET_YES: M3 Processor has been reset
6	EMULATION_M3_RST	R/W	0h	M3 Processor has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module 0 - RESET_NO: No reset 1 - RESET_YES: M3 Processor has been reset
5	WKUP_M3_LRST	R/W	0h	M3 Processor has been reset 0 - RESET_NO: No reset 1 - RESET_YES: M3 Processor has been reset
4-0	Reserved	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.1.5.6.3 PM_WKUP_PWRSTCTRL Register (offset = 4h)

Controls power state of WKUP power domain

PM_WKUP_PWRSTCTRL is shown in Figure 7-22 and described in Table 7-28.

Figure 7-22. PM_WKUP_PWRSTCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-28. PM_WKUP_PWRSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	Reserved	R	0h	
29	Reserved	R	0h	
28	Reserved	R	0h	
27	Reserved	R	0h	
26-25	Reserved	R	0h	
24-5	Reserved	R	0h	
4	LowPowerStateChange	R/W	0h	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0 - DIS: Do not request a low power state change. 1 - EN: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.
3	LogicRETState	R/W	1h	Logic state when power domain is RETENTION 0 - logic_off: Only retention registers are retained and remaing logic is off when the domain is in RETENTION state. 1 - logic_ret: Whole logic is retained when domain is in RETENTION state.
2	Reserved	R	0h	
1-0	Reserved	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.1.5.6.4 PM_WKUP_PWRSTST Register (offset = 8h)

This register provides a status on the current WKUP power domain state. [warm reset insensitive]

PM_WKUP_PWRSTST is shown in Figure 7-23 and described in Table 7-29.

Figure 7-23. PM_WKUP_PWRSTST Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-29. PM_WKUP_PWRSTST Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	Reserved	R	0h	
22-21	Reserved	R	0h	
20	InTransition	R	0h	Domain transition status 0 - No: No on-going transition on power domain 1 - Ongoing: Power domain transition is in progress.
19	Reserved	R	0h	
18-17	Debugss_mem_statest	R	3h	WKUP domain memory state status 0 - Mem_off: Memory is OFF 2 - Reserved: Reserved 3 - Mem_on: Memory is ON
16-3	Reserved	R	0h	
2	LogicStateSt	R	1h	Logic state status 0 - OFF: Logic in domain is OFF 1 - ON: Logic in domain is ON
1-0	Reserved	R	0h	

7.2 Clock Management

7.2.1 Introduction

7.2.2 Integration

7.2.3 Functional Description

7.2.4 Use Cases

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5 Registers

7.2.5.0.5 CM_CEFUSE Registers

Table 7-30 lists the memory-mapped registers for the CM_CEFUSE. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 7-30 should be considered as reserved locations and the register contents should not be modified.

Table 7-30. CM_CEFUSE REGISTERS

Offset	Acronym	Register Description	Section
0h	CM_CEFUSE_CLKSTCTRL	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.	Section 7.2.5.0.5.1
20h	CM_CEFUSE_CEFUSE_CLKCTRL	This register manages the CEFUSE clocks.	Section 7.2.5.0.5.2

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.5.1 CM_CEFUSE_CLKSTCTRL Register (offset = 0h)

This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

CM_CEFUSE_CLKSTCTRL is shown in Figure 7-24 and described in Table 7-31.

Figure 7-24. CM_CEFUSE_CLKSTCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-31. CM_CEFUSE_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	Reserved	R	0h	
9	CLKACTIVITY_CUST_EFUSE_SYS_CLK	R	0h	This field indicates the state of the Cust_Efuse_SYSCLK clock input of the domain. [warm reset insensitive] 0 - Inact: Corresponding clock is definitely gated 1 - Act: Corresponding clock is running or gating/ungating transition is on-going
8	CLKACTIVITY_L4_CEFUSE_GICLK	R	0h	This field indicates the state of the L4_CEFUSE_GCLK clock input of the domain. [warm reset insensitive] 0 - Inact: Corresponding clock is definitely gated 1 - Act: Corresponding clock is running or gating/ungating transition is on-going
7-2	Reserved	R	0h	
1-0	CLKTRCTRL	R/W	2h	Controls the clock state transition of the clock domain in customer efuse power domain. 0 - NO_SLEEP: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1 - SW_SLEEP: SW_SLEEP: Start a software forced sleep transition on the domain. 2 - SW_WKUP: SW_WKUP: Start a software forced wake-up transition on the domain. 3 - HW_AUTO: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.5.2 CM_CEFUSE_CEFUSE_CLKCTRL Register (offset = 20h)

This register manages the CEFUSE clocks.

CM_CEFUSE_CEFUSE_CLKCTRL is shown in Figure 7-25 and described in Table 7-32.

Figure 7-25. CM_CEFUSE_CEFUSE_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-32. CM_CEFUSE_CEFUSE_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. [warm reset insensitive] 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.6 CM_DEVICE Registers

Table 7-33 lists the memory-mapped registers for the CM_DEVICE. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 7-33 should be considered as reserved locations and the register contents should not be modified.

Table 7-33. CM_DEVICE REGISTERS

Offset	Acronym	Register Description	Section
0h	CM_CLKOUT_CTRL	This register provides the control over CLKOUT1 and CLKOUT2 output	Section 7.2.5.0.6.1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.6.1 CM_CLKOUT_CTRL Register (offset = 0h)

This register provides the control over CLKOUT1 and CLKOUT2 output

CM_CLKOUT_CTRL is shown in Figure 7-26 and described in Table 7-34.

Figure 7-26. CM_CLKOUT_CTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-34. CM_CLKOUT_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	Reserved	R	0h	
7	CLKOUT2EN	R/W	0h	This bit controls the external clock activity 0 - DIS: SYS_CLKOUT2 is disabled 1 - EN: SYS_CLKOUT2 is enabled
6	Reserved	R	0h	
5-3	CLKOUT2DIV	R/W	0h	THis field controls the external clock divison factor 0 - DIV1: SYS_CLKOUT2/1 1 - DIV2: SYS_CLKOUT2/2 2 - DIV3: SYS_CLKOUT2/3 3 - DIV4: SYS_CLKOUT2/4 4 - DIV5: SYS_CLKOUT2/5 5 - DIV6: SYS_CLKOUT2/6 6 - DIV7: SYS_CLKOUT2/7 7 - DIV8: SYS_CLKOUT2/8
2-0	CLKOUT2SOURCE	R/W	0h	This field selects the external output clock source 0 - SEL0: Select 32KHz Oscillator O/P 1 - SEL1: Select L3 Clock 2 - SEL2: Select DDR PHY Clock 3 - SEL4: Select 192Mhz clock from PER PLL 4 - SEL5: Select LCD Pixel Clock

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.7 CM_DPLL Registers

Table 7-35 lists the memory-mapped registers for the CM_DPLL. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 7-35 should be considered as reserved locations and the register contents should not be modified.

Table 7-35. CM_DPLL REGISTERS

Offset	Acronym	Register Description	Section
4h + (0h)	CLKSEL_TIMER7_CLK	Selects the Mux select line for TIMER7 clock [warm reset insensitive]	Section 7.2.5.0.7.1
4h + (Ch)	CLKSEL_TIMER4_CLK	Selects the Mux select line for TIMER4 clock [warm reset insensitive]	Section 7.2.5.0.7.2
4h + (10h)	CM_MAC_CLKSEL	Selects the clock divide ration for MII clock [warm reset insensitive]	Section 7.2.5.0.7.3
4h + (14h)	CLKSEL_TIMER5_CLK	Selects the Mux select line for TIMER5 clock [warm reset insensitive]	Section 7.2.5.0.7.4
4h + (18h)	CLKSEL_TIMER6_CLK	Selects the Mux select line for TIMER6 clock [warm reset insensitive]	Section 7.2.5.0.7.5
4h + (1Ch)	CM_CPTS_RFT_CLKSEL	Selects the Mux select line for CPTS RFT clock [warm reset insensitive]	Section 7.2.5.0.7.6
4h + (24h)	CLKSEL_TIMER1MS_CLK	Selects the Mux select line for TIMER1 clock [warm reset insensitive]	Section 7.2.5.0.7.7
4h + (4h)	CLKSEL_TIMER2_CLK	Selects the Mux select line for TIMER2 clock [warm reset insensitive]	Section 7.2.5.0.7.8
4h + (28h)	CLKSEL_GFX_FCLK	Selects the divider value for GFX clock [warm reset insensitive]	Section 7.2.5.0.7.9
4h + (2Ch)	CLKSEL_ICSS_OCP_CLK	Controls the Mux select line for ICSS OCP clock [warm reset insensitive]	Section 7.2.5.0.7.10
4h + (30h)	CLKSEL_LCDC_PIXEL_CLK	Controls the Mux select line for LCDC PIXEL clock [warm reset insensitive]	Section 7.2.5.0.7.11
4h + (34h)	CLKSEL_WDT1_CLK	Selects the Mux select line for Watchdog1 clock [warm reset insensitive]	Section 7.2.5.0.7.12
4h + (38h)	CLKSEL_GPIO0_DBCLK	Selects the Mux select line for GPIO0 debounce clock [warm reset insensitive]	Section 7.2.5.0.7.13
4h + (8h)	CLKSEL_TIMER3_CLK	Selects the Mux select line for TIMER3 clock [warm reset insensitive]	Section 7.2.5.0.7.14

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.7.1 CLKSEL_TIMER7_CLK Register (offset = 4h + (0h))

Selects the Mux select line for TIMER7 clock [warm reset insensitive]

CLKSEL_TIMER7_CLK is shown in Figure 7-27 and described in Table 7-36.

Figure 7-27. CLKSEL_TIMER7_CLK Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-36. CLKSEL_TIMER7_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	Reserved	R	0h	
1-0	CLKSEL	R/W	1h	Selects the Mux select line for TIMER7 clock [warm reset insensitive] 0 - SEL1: Select TCLKIN clock 1 - SEL2: Select CLK_M_OSC clock 2 - SEL3: Select CLK_32KHZ clock 3 - SEL4: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.7.2 CLKSEL_TIMER4_CLK Register (offset = 4h + (Ch))

Selects the Mux select line for TIMER4 clock [warm reset insensitive]

CLKSEL_TIMER4_CLK is shown in Figure 7-28 and described in Table 7-37.

Figure 7-28. CLKSEL_TIMER4_CLK Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-37. CLKSEL_TIMER4_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	Reserved	R	0h	
1-0	CLKSEL	R/W	1h	Selects the Mux select line for TIMER4 clock [warm reset insensitive] 0 - SEL1: Select TCLKIN clock 1 - SEL2: Select CLK_M_OSC clock 2 - SEL3: Select CLK_32KHZ clock 3 - SEL4: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.7.3 CM_MAC_CLKSEL Register (offset = 4h + (10h))

Selects the clock divide ration for MII clock [warm reset insensitive]

CM_MAC_CLKSEL is shown in Figure 7-29 and described in Table 7-38.

Figure 7-29. CM_MAC_CLKSEL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-38. CM_MAC_CLKSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	Reserved	R	0h	
2	MII_CLK_SEL	R/W	1h	MII Clock Divider Selection. This bit is warm reset insensitive when CPSW RESET_ISO is enabled 0 - SEL0: Selects 1/2 divider of SYSCLK2 1 - SEL1: Selects 1/5 divide ratio of SYSCLK2
1-0	Reserved	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.7.4 CLKSEL_TIMER5_CLK Register (offset = 4h + (14h))

Selects the Mux select line for TIMER5 clock [warm reset insensitive]

CLKSEL_TIMER5_CLK is shown in Figure 7-30 and described in Table 7-39.

Figure 7-30. CLKSEL_TIMER5_CLK Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-39. CLKSEL_TIMER5_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	Reserved	R	0h	
1-0	CLKSEL	R/W	1h	Selects the Mux select line for TIMER5 clock [warm reset insensitive] 0 - SEL1: Select TCLKIN clock 1 - SEL2: Select CLK_M_OSC clock 2 - SEL3: Select CLK_32KHZ clock 3 - SEL4: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.7.5 CLKSEL_TIMER6_CLK Register (offset = 4h + (18h))

Selects the Mux select line for TIMER6 clock [warm reset insensitive]

CLKSEL_TIMER6_CLK is shown in Figure 7-31 and described in Table 7-40.

Figure 7-31. CLKSEL_TIMER6_CLK Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-40. CLKSEL_TIMER6_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	Reserved	R	0h	
1-0	CLKSEL	R/W	1h	Selects the Mux select line for TIMER6 clock [warm reset insensitive] 0 - SEL1: Select TCLKIN clock 1 - SEL2: Select CLK_M_OSC clock 2 - SEL3: Select CLK_32KHZ clock 3 - SEL4: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.7.6 CM_CPTS_RFT_CLKSEL Register (offset = 4h + (1Ch))

Selects the Mux select line for CPTS RFT clock [warm reset insensitive]

CM_CPTS_RFT_CLKSEL is shown in Figure 7-32 and described in Table 7-41.

Figure 7-32. CM_CPTS_RFT_CLKSEL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-41. CM_CPTS_RFT_CLKSEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	Reserved	R	0h	
2	Reserved	R	0h	
1	Reserved	R	0h	
0	CLKSEL	R/W	0h	Selects the Mux select line for cpgmac rft clock [warm reset insensitive] 0 - SEL1: Select McBSP audio clock to be SYSCLK20 1 - SEL2: Select McBSP audio clock to be SYSCLK21

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.7.7 CLKSEL_TIMER1MS_CLK Register (offset = 4h + (24h))

Selects the Mux select line for TIMER1 clock [warm reset insensitive]

CLKSEL_TIMER1MS_CLK is shown in Figure 7-33 and described in Table 7-42.

Figure 7-33. CLKSEL_TIMER1MS_CLK Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-42. CLKSEL_TIMER1MS_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	Reserved	R	0h	
2-0	CLKSEL	R/W	0h	Selects the Mux select line for DMTIMER_1MS clock [warm reset insensitive] 0 - SEL1: Select CLK_M_OSC clock 1 - SEL2: Select CLK_32KHZ clock 2 - SEL3: Select TCLKIN clock 3 - SEL4: Select CLK_RC32K clock 4 - SEL5: Selects the CLK_32768 from 32KHz Crystal Osc

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.7.8 CLKSEL_TIMER2_CLK Register (offset = 4h + (4h))

Selects the Mux select line for TIMER2 clock [warm reset insensitive]

CLKSEL_TIMER2_CLK is shown in Figure 7-34 and described in Table 7-43.

Figure 7-34. CLKSEL_TIMER2_CLK Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-43. CLKSEL_TIMER2_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	Reserved	R	0h	
1-0	CLKSEL	R/W	1h	Selects the Mux select line for TIMER2 clock [warm reset insensitive] 0 - SEL1: Select TCLKIN clock 1 - SEL2: Select CLK_M_OSC clock 2 - SEL3: Select CLK_32KHZ clock 3 - SEL4: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.7.9 CLKSEL_GFX_FCLK Register (offset = 4h + (28h))

Selects the divider value for GFX clock [warm reset insensitive]

CLKSEL_GFX_FCLK is shown in Figure 7-35 and described in Table 7-44.

Figure 7-35. CLKSEL_GFX_FCLK Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-44. CLKSEL_GFX_FCLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	Reserved	R	0h	
1	CLKSEL_GFX_FCLK	R/W	0h	Selects the clock on gfx fclk [warm reset insensitive] 0 - SEL0: SGX FCLK is from CORE PLL (same as L3 clock) 1 - SEL1: SGX FCLK is from PER PLL (192 MHz clock)
0	CLKDIV_SEL_GFX_FCLK	R/W	0h	Selects the divider value on gfx fclk [warm reset insensitive] 0 - DIV1: SGX FCLK is same as L3 Clock or 192MHz Clock 1 - DIV2: SGX FCLK is L3 clock/2 or 192Mhz/2

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.7.10 CLKSEL_ICSS_OCP_CLK Register (offset = 4h + (2Ch))

Controls the Mux select line for ICSS OCP clock [warm reset insensitive]

CLKSEL_ICSS_OCP_CLK is shown in Figure 7-36 and described in Table 7-45.

Figure 7-36. CLKSEL_ICSS_OCP_CLK Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-45. CLKSEL_ICSS_OCP_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R	0h	
0	CLKSEL	R/W	0h	Controls Mux Select of ICSS OCP clock mux 0 - SEL1: Select L3F clock as OCP Clock of ICSS 1 - SEL2: Select DISP DPLL clock as OCP clock of ICSS

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.7.11 CLKSEL_LCDC_PIXEL_CLK Register (offset = 4h + (30h))

Controls the Mux select line for LCDC PIXEL clock [warm reset insensitive]

CLKSEL_LCDC_PIXEL_CLK is shown in Figure 7-37 and described in Table 7-46.

Figure 7-37. CLKSEL_LCDC_PIXEL_CLK Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-46. CLKSEL_LCDC_PIXEL_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	Reserved	R	0h	
1-0	CLKSEL	R/W	0h	Controls the Mux Select of LCDC PIXEL clock 0 - SEL1: Select DISP PLL CLKOUTM2 1 - SEL2: Select CORE PLL CLKOUTM5 2 - SEL3: Select PER PLL CLKOUTM2 3 - SEL4: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.7.12 CLKSEL_WDT1_CLK Register (offset = 4h + (34h))

Selects the Mux select line for Watchdog1 clock [warm reset insensitive]

CLKSEL_WDT1_CLK is shown in Figure 7-38 and described in Table 7-47.

Figure 7-38. CLKSEL_WDT1_CLK Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-47. CLKSEL_WDT1_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R	0h	
0	CLKSEL	R/W	1h	Selects the Mux select line for WDT1 clock [warm reset insensitive] 0 - SEL1: Select 32KHZ clock from RC Oscillator 1 - SEL2: Select 32KHZ from 32K Clock divider

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.7.13 CLKSEL_GPIO0_DBCLK Register (offset = 4h + (38h))

Selects the Mux select line for GPIO0 debounce clock [warm reset insensitive]

CLKSEL_GPIO0_DBCLK is shown in Figure 7-39 and described in Table 7-48.

Figure 7-39. CLKSEL_GPIO0_DBCLK Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-48. CLKSEL_GPIO0_DBCLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	Reserved	R	0h	
1-0	CLKSEL	R/W	0h	Selects the Mux select line for GPIO0 debounce clock [warm reset insensitive] 0 - SEL1: Select 32KHZ clock from RC Oscillator 1 - SEL2: Select 32KHZ from 32K Crystal Oscillator 2 - SEL3: Select 32KHz from Clock Divider

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.7.14 CLKSEL_TIMER3_CLK Register (offset = 4h + (8h))

Selects the Mux select line for TIMER3 clock [warm reset insensitive]

CLKSEL_TIMER3_CLK is shown in Figure 7-40 and described in Table 7-49.

Figure 7-40. CLKSEL_TIMER3_CLK Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-49. CLKSEL_TIMER3_CLK Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	Reserved	R	0h	
1-0	CLKSEL	R/W	1h	Selects the Mux select line for TIMER3 clock [warm reset insensitive] 0 - SEL1: Select TCLKIN clock 1 - SEL2: Select CLK_M_OSC clock 2 - SEL3: Select CLK_32KHZ clock 3 - SEL4: Reserved

7.2.5.0.8 CM_GFX Registers

Table 7-50 lists the memory-mapped registers for the CM_GFX. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 7-50 should be considered as reserved locations and the register contents should not be modified.

Table 7-50. CM_GFX REGISTERS

Offset	Acronym	Register Description	Section
0h	CM_GFX_L3_CLKSTCTRL	This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.	Section 7.2.5.0.8.1
Ch	CM_GFX_L4LS_GFX_CLKSTCTRL	This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.	Section 7.2.5.0.8.2
10h	CM_GFX_MMUCFG_CLKCTRL	This register manages the MMU CFG clocks.	Section 7.2.5.0.8.3
14h	CM_GFX_MMUDATA_CLKCTRL	This register manages the MMU clocks.	Section 7.2.5.0.8.4
4h	CM_GFX_GFX_CLKCTRL	This register manages the GFX clocks.	Section 7.2.5.0.8.5
8h	CM_GFX_BITBLT_CLKCTRL	This register manages the BITBLIT clocks.	Section 7.2.5.0.8.6

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.8.1 CM_GFX_L3_CLKSTCTRL Register (offset = 0h) [reset = 0h]

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

CM_GFX_L3_CLKSTCTRL is shown in Figure 7-41 and described in Table 7-51.

Figure 7-41. CM_GFX_L3_CLKSTCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-51. CM_GFX_L3_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	R	XXXh	
25-10	Reserved	R	XXXh	
9	CLKACTIVITY_GFX_FCLK	R	XXXh	This field indicates the state of the GFX_GCLK clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
8	CLKACTIVITY_GFX_L3_GCLK	R	XXXh	This field indicates the state of the GFX_L3_GCLK clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
7-2	Reserved	R	XXXh	
1-0	CLKTRCTRL	R/W	XXXh	Controls the clock state transition of the GFX clock domain in GFX power domain. 0 - NO_SLEEP: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1 - SW_SLEEP: SW_SLEEP: Start a software forced sleep transition on the domain. 2 - SW_WKUP: SW_WKUP: Start a software forced wake-up transition on the domain. 3 - HW_AUTO: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.8.2 CM_GFX_L4LS_GFX_CLKSTCTRL Register (offset = Ch) [reset = 0h]

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

CM_GFX_L4LS_GFX_CLKSTCTRL is shown in Figure 7-42 and described in Table 7-52.

Figure 7-42. CM_GFX_L4LS_GFX_CLKSTCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-52. CM_GFX_L4LS_GFX_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R	XXXh	
8	CLKACTIVITY_L4LS_GFX_GCLK	R	XXXh	This field indicates the state of the L4_LS clock in the domain. 0 - Inact 1 - Act
7-2	Reserved	R	XXXh	
1-0	CLKTRCTRL	R/W	XXXh	Controls the clock state transition of the L4LS clock domain in GFX power domain. 0 - NO_SLEEP: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1 - SW_SLEEP: SW_SLEEP: Start a software forced sleep transition on the domain. 2 - SW_WKUP: SW_WKUP: Start a software forced wake-up transition on the domain. 3 - HW_AUTO: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.8.3 CM_GFX_MMUCFG_CLKCTRL Register (offset = 10h) [reset = 0h]

This register manages the MMU CFG clocks.

CM_GFX_MMUCFG_CLKCTRL is shown in Figure 7-43 and described in Table 7-53.

Figure 7-43. CM_GFX_MMUCFG_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-53. CM_GFX_MMUCFG_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	XXXh	
17-16	IDLEST	R	XXXh	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	XXXh	
1-0	MODULEMODE	R/W	XXXh	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.8.4 CM_GFX_MMUDATA_CLKCTRL Register (offset = 14h) [reset = 0h]

This register manages the MMU clocks.

CM_GFX_MMUDATA_CLKCTRL is shown in Figure 7-44 and described in Table 7-54.

Figure 7-44. CM_GFX_MMUDATA_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-54. CM_GFX_MMUDATA_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	XXXh	
17-16	IDLEST	R	XXXh	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	XXXh	
1-0	MODULEMODE	R/W	XXXh	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.8.5 CM_GFX_GFX_CLKCTRL Register (offset = 4h) [reset = 0h]

This register manages the GFX clocks.

CM_GFX_GFX_CLKCTRL is shown in Figure 7-45 and described in Table 7-55.

Figure 7-45. CM_GFX_GFX_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-55. CM_GFX_GFX_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	XXXh	
18	STBYST	R	XXXh	Module standby status. 0 - Func: Module is functional (not in standby) 1 - Standby: Module is in standby
17-16	IDLEST	R	XXXh	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	XXXh	
1-0	MODULEMODE	R/W	XXXh	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.8.6 CM_GFX_BITBLT_CLKCTRL Register (offset = 8h) [reset = 0h]

This register manages the BITBLIT clocks.

CM_GFX_BITBLT_CLKCTRL is shown in Figure 7-46 and described in Table 7-56.

Figure 7-46. CM_GFX_BITBLT_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-56. CM_GFX_BITBLT_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	XXXh	
18	STBYST	R	XXXh	Module standby status. 0 - Func: Module is functional (not in standby) 1 - Standby: Module is in standby
17-16	IDLEST	R	XXXh	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	XXXh	
1-0	MODULEMODE	R/W	XXXh	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

7.2.5.0.9 CM_MPU Registers

Table 7-57 lists the memory-mapped registers for the CM_MPU. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 7-57 should be considered as reserved locations and the register contents should not be modified.

Table 7-57. CM_MPU REGISTERS

Offset	Acronym	Register Description	Section
0h	CM_MPU_CLKSTCTRL	This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.	Section 7.2.5.0.9.1
4h	CM_MPU_MPU_CLKCTRL	This register manages the MPU clocks.	Section 7.2.5.0.9.2

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.9.1 CM_MPU_CLKSTCTRL Register (offset = 0h) [reset = 0h]

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

CM_MPU_CLKSTCTRL is shown in Figure 7-47 and described in Table 7-58.

Figure 7-47. CM_MPU_CLKSTCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-58. CM_MPU_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	Reserved	R	XXXh	
2	CLKACTIVITY_MPU_CLK	R	XXXh	This field indicates the state of the MPU Clock 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
1-0	CLKTRCTRL	R/W	XXXh	Controls the clock state transition of the MPU clock domains. 0 - NO_SLEEP: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1 - SW_SLEEP: SW_SLEEP: Start a software forced sleep transition on the domain. 2 - SW_WKUP: SW_WKUP: Start a software forced wake-up transition on the domain. 3 - HW_AUTO: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.9.2 CM_MPU_MPU_CLKCTRL Register (offset = 4h) [reset = 0h]

This register manages the MPU clocks.

CM_MPU_MPU_CLKCTRL is shown in Figure 7-48 and described in Table 7-59.

Figure 7-48. CM_MPU_MPU_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-59. CM_MPU_MPU_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	XXXh	
18	STBYST	R	XXXh	Module standby status. 0 - Func: Module is functional (not in standby) 1 - Standby: Module is in standby
17-16	IDLEST	R	XXXh	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	XXXh	
1-0	MODULEMODE	R/W	XXXh	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10 CM_PER Registers

Table 7-60 lists the memory-mapped registers for the CM_PER. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 7-60 should be considered as reserved locations and the register contents should not be modified.

Table 7-60. CM_PER REGISTERS

Offset	Acronym	Register Description	Section
0h	CM_PER_L4LS_CLKSTCTRL	This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-PER and ON-INPER states. It also hold one status bit per clock input of the domain.	Section 7.2.5.0.10.1
64h	CM_PER_L4FW_CLKCTRL	This register manages the L4FW clocks.	Section 7.2.5.0.10.2
68h	CM_PER_MCASP1_CLKCTRL	This register manages the MCASP1 clocks.	Section 7.2.5.0.10.3
6Ch	CM_PER_UART1_CLKCTRL	This register manages the IART1 clocks.	Section 7.2.5.0.10.4
70h	CM_PER_UART2_CLKCTRL	This register manages the UART2 clocks.	Section 7.2.5.0.10.5
74h	CM_PER_UART3_CLKCTRL	This register manages the UART3 clocks.	Section 7.2.5.0.10.6
Ch	CM_PER_L3_CLKSTCTRL	This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.	Section 7.2.5.0.10.7
78h	CM_PER_UART4_CLKCTRL	This register manages the UART4 clocks.	Section 7.2.5.0.10.8
7Ch	CM_PER_TIMER7_CLKCTRL	This register manages the TIMER7 clocks.	Section 7.2.5.0.10.9
80h	CM_PER_TIMER2_CLKCTRL	This register manages the TIMER2 clocks.	Section 7.2.5.0.10.10
84h	CM_PER_TIMER3_CLKCTRL	This register manages the TIMER3 clocks.	Section 7.2.5.0.10.11
88h	CM_PER_TIMER4_CLKCTRL	This register manages the TIMER4 clocks.	Section 7.2.5.0.10.12
8Ch	CM_PER_MCASP2_CLKCTRL	This register manages the MCASP2 clocks.	Section 7.2.5.0.10.13
90h	CM_PER_RNG_CLKCTRL	This register manages the RNG clocks.	Section 7.2.5.0.10.14
94h	CM_PER_AES0_CLKCTRL	This register manages the AES0 clocks.	Section 7.2.5.0.10.15
98h	CM_PER_AES1_CLKCTRL	This register manages the AES1 clocks.	Section 7.2.5.0.10.16
9Ch	CM_PER_DES_CLKCTRL	This register manages the DES clocks.	Section 7.2.5.0.10.17
10h	Reserved		Section 7.2.5.0.10.18
A0h	CM_PER_SHA0_CLKCTRL	This register manages the SHA0 clocks.	Section 7.2.5.0.10.19
A4h	CM_PER_PKA_CLKCTRL	This register manages the PKA clocks.	Section 7.2.5.0.10.20
A8h	CM_PER_GPIO6_CLKCTRL	This register manages the GPIO6 clocks.	Section 7.2.5.0.10.21
ACh	CM_PER_GPIO1_CLKCTRL	This register manages the GPIO1 clocks.	Section 7.2.5.0.10.22
B0h	CM_PER_GPIO2_CLKCTRL	This register manages the GPIO2 clocks.	Section 7.2.5.0.10.23
B4h	CM_PER_GPIO3_CLKCTRL	This register manages the GPIO3 clocks.	Section 7.2.5.0.10.24
B8h	CM_PER_GPIO4_CLKCTRL	This register manages the GPIO4 clocks.	Section 7.2.5.0.10.25
BCh	CM_PER_TPCC_CLKCTRL	This register manages the TPCC clocks.	Section 7.2.5.0.10.26

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 7-60. CM_PER REGISTERS (continued)

Offset	Acronym	Register Description	Section
C0h	CM_PER_DCAN0_CLKCTRL	This register manages the DCAN0 clocks.	Section 7.2.5.0.10.27
C4h	CM_PER_DCAN1_CLKCTRL	This register manages the DCAN1 clocks.	Section 7.2.5.0.10.28
14h	CM_PER_CPGMAC0_CLKCTRL	This register manages the CPSW clocks.	Section 7.2.5.0.10.29
C8h	CM_PER_SPARE_CLKCTRL	This register is reserved	Section 7.2.5.0.10.30
CCh	CM_PER_EPWMSS1_CLKCTRL	This register manages the PWMSS1 clocks.	Section 7.2.5.0.10.31
D0h	CM_PER_EMIF_FW_CLKCTRL	This register manages the EMIF Firewall clocks.	Section 7.2.5.0.10.32
D4h	CM_PER_EPWMSS0_CLKCTRL	This register manages the PWMSS0 clocks.	Section 7.2.5.0.10.33
D8h	CM_PER_EPWMSS2_CLKCTRL	This register manages the PWMSS2 clocks.	Section 7.2.5.0.10.34
DCh	CM_PER_L3_INSTR_CLKCTRL	This register manages the L3 INSTR clocks.	Section 7.2.5.0.10.35
E0h	CM_PER_L3_CLKCTRL	This register manages the L3 Interconnect clocks.	Section 7.2.5.0.10.36
E4h	CM_PER_IEEE5000_CLKCTRL	This register manages the IEEE1500 clocks.	Section 7.2.5.0.10.37
E8h	CM_PER_ICSS_CLKCTRL	This register manages the ICSS clocks.	Section 7.2.5.0.10.38
ECh	CM_PER_TIMER5_CLKCTRL	This register manages the TIMER5 clocks.	Section 7.2.5.0.10.39
18h	CM_PER_LCDC_CLKCTRL	This register manages the LCD clocks.	Section 7.2.5.0.10.40
F0h	CM_PER_TIMER6_CLKCTRL	This register manages the TIMER6 clocks.	Section 7.2.5.0.10.41
F4h	CM_PER_MMC1_CLKCTRL	This register manages the MMC1 clocks.	Section 7.2.5.0.10.42
F8h	CM_PER_MMC2_CLKCTRL	This register manages the MMC2 clocks.	Section 7.2.5.0.10.43
FCh	CM_PER_TPTC1_CLKCTRL	This register manages the TPTC1 clocks.	Section 7.2.5.0.10.44
100h	CM_PER_TPTC2_CLKCTRL	This register manages the TPTC2 clocks.	Section 7.2.5.0.10.45
104h	CM_PER_GPIO5_CLKCTRL	This register manages the GPIO5 clocks.	Section 7.2.5.0.10.46
10Ch	CM_PER_SPINLOCK_CLKCTRL	This register manages the SPINLOCK clocks.	Section 7.2.5.0.10.47
110h	CM_PER_MAILBOX0_CLKCTRL	This register manages the MAILBOX0 clocks.	Section 7.2.5.0.10.48
1Ch	CM_PER_USB0_CLKCTRL	This register manages the USB clocks.	Section 7.2.5.0.10.49
11Ch	CM_PER_L4HS_CLKSTCTRL	This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.	Section 7.2.5.0.10.50
120h	CM_PER_L4HS_CLKCTRL	This register manages the L4 Fast clocks.	Section 7.2.5.0.10.51
124h	CM_PER_MSTR_EXPS_CLKCTRL	This register manages the Master Expansion Port clocks.	Section 7.2.5.0.10.52
128h	CM_PER_SLV_EXPS_CLKCTRL	This register manages the Slave Expansion Port clocks.	Section 7.2.5.0.10.53

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 7-60. CM_PER REGISTERS (continued)

Offset	Acronym	Register Description	Section
12Ch	CM_PER_OCPWP_L3_CLKSTCTRL	This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.	Section 7.2.5.0.10.54
130h	CM_PER_OCPWP_CLKCTRL	This register manages the OCPWP clocks.	Section 7.2.5.0.10.55
134h	CM_PER_MAILBOX1_CLKCTRL	This register manages the Mailbox1 clocks.	Section 7.2.5.0.10.56
138h	CM_PER_SPARE0_CLKCTRL	This register manages the Spare0 clocks.	Section 7.2.5.0.10.57
13Ch	CM_PER_SPARE1_CLKCTRL	This register manages the Spare1 clocks.	Section 7.2.5.0.10.58
20h	CM_PER_MLB_CLKCTRL	This register manages the MLB clocks.	Section 7.2.5.0.10.59
140h	CM_PER_ICSS_CLKSTCTRL	This register enables the clock domain state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.	Section 7.2.5.0.10.60
144h	CM_PER_CPSW_CLKSTCTRL	This register enables the clock domain state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.	Section 7.2.5.0.10.61
148h	CM_PER_LCDC_CLKSTCTRL	This register enables the clock domain state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.	Section 7.2.5.0.10.62
14Ch	CM_PER_CLKDIV32K_CLKCTRL	This register manages the CLKDIV32K clocks.	Section 7.2.5.0.10.63
150h	CM_PER_CLK_24MHZ_CLKSTCTRL	This register enables the clock domain state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.	Section 7.2.5.0.10.64
24h	CM_PER_TPTC0_CLKCTRL	This register manages the TPTC clocks.	Section 7.2.5.0.10.65
4h	CM_PER_L3S_CLKSTCTRL	This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.	Section 7.2.5.0.10.66
28h	CM_PER_EMIF_CLKCTRL	This register manages the EMIF clocks.	Section 7.2.5.0.10.67
2Ch	CM_PER_OCMCRAM_CLKCTRL	This register manages the OCMC clocks.	Section 7.2.5.0.10.68
30h	CM_PER_GPMC_CLKCTRL	This register manages the GPMC clocks.	Section 7.2.5.0.10.69
34h	CM_PER_MCASP0_CLKCTRL	This register manages the MCASP0 clocks.	Section 7.2.5.0.10.70
38h	CM_PER_UART5_CLKCTRL	This register manages the UART5 clocks.	Section 7.2.5.0.10.71
3Ch	CM_PER_MMC0_CLKCTRL	This register manages the MMC0 clocks.	Section 7.2.5.0.10.72
40h	CM_PER_ELM_CLKCTRL	This register manages the ELM clocks.	Section 7.2.5.0.10.73
44h	CM_PER_I2C2_CLKCTRL	This register manages the I2C2 clocks.	Section 7.2.5.0.10.74
48h	CM_PER_I2C1_CLKCTRL	This register manages the I2C1 clocks.	Section 7.2.5.0.10.75

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 7-60. CM_PER REGISTERS (continued)

Offset	Acronym	Register Description	Section
4Ch	CM_PER_SPI0_CLKCTRL	This register manages the SPI0 clocks.	Section 7.2.5.0.10.76
8h	CM_PER_L4FW_CLKSTCTRL	This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.	Section 7.2.5.0.10.77
50h	CM_PER_SPI1_CLKCTRL	This register manages the SPI1 clocks.	Section 7.2.5.0.10.78
54h	CM_PER_SPI2_CLKCTRL	This register manages the SPI2 clocks.	Section 7.2.5.0.10.79
58h	CM_PER_SPI3_CLKCTRL	This register manages the SPI3 clocks.	Section 7.2.5.0.10.80
60h	CM_PER_L4LS_CLKCTRL	This register manages the L4LS clocks.	Section 7.2.5.0.10.81

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.1 CM_PER_L4LS_CLKSTCTRL Register (offset = 0h)

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-PER and ON-INPER states. It also hold one status bit per clock input of the domain.

CM_PER_L4LS_CLKSTCTRL is shown in Figure 7-49 and described in Table 7-61.

Figure 7-49. CM_PER_L4LS_CLKSTCTRL Register

31	30	29	28	27	26	25	24
Reserved			CLKACTIVITY_TIMER6_GCLK	CLKACTIVITY_TIMER5_GCLK	CLKACTIVITY_GPIO_5_GDBCLK	CLKACTIVITY_SPI_GCLK	CLKACTIVITY_I2C_FCLK
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h

23	22	21	20	19	18	17	16
Reserved	CLKACTIVITY_GPIO_4_GDBCLK	CLKACTIVITY_GPIO_3_GDBCLK	CLKACTIVITY_GPIO_2_GDBCLK	CLKACTIVITY_GPIO_1_GDBCLK	CLKACTIVITY_GPIO_6_GDBCLK	CLKACTIVITY_LCDC_GCLK	CLKACTIVITY_TIMER4_GCLK
R-0h	R-0h	R-0h	R-0h	R-1h	R-1h	R-0h	R-0h

15	14	13	12	11	10	9	8
CLKACTIVITY_TIMER3_GCLK	CLKACTIVITY_TIMER2_GCLK	CLKACTIVITY_TIMER7_GCLK	Reserved	CLKACTIVITY_CAN_CLK	CLKACTIVITY_UART_GFCLK	Reserved	CLKACTIVITY_L4LS_GCLK
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-1h

7	6	5	4	3	2	1	0
Reserved						CLKTRCTRL	
R-0h						R/W-2h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-61. CM_PER_L4LS_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-29	Reserved	R	0h	
28	CLKACTIVITY_TIMER6_GCLK	R	0h	This field indicates the state of the TIMER6 CLKTIMER clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
27	CLKACTIVITY_TIMER5_GCLK	R	0h	This field indicates the state of the TIMER5 CLKTIMER clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
26	CLKACTIVITY_GPIO_5_GDBCLK	R	0h	This field indicates the state of the GPIO DBCLK clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
25	CLKACTIVITY_SPI_GCLK	R	0h	This field indicates the state of the SPI_GCLK clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
24	CLKACTIVITY_I2C_FCLK	R	0h	This field indicates the state of the I2C _FCLK clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
23	Reserved	R	0h	
22	CLKACTIVITY_GPIO_4_GDBCLK	R	0h	This field indicates the state of the GPIO4_GDBCLK clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
21	CLKACTIVITY_GPIO_3_GDBCLK	R	0h	This field indicates the state of the GPIO3_GDBCLK clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
20	CLKACTIVITY_GPIO_2_GDBCLK	R	0h	This field indicates the state of the GPIO2_ GDBCLK clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 7-61. CM_PER_L4LS_CLKSTCTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
19	CLKACTIVITY_GPIO_1_GDBCLK	R	1h	This field indicates the state of the GPIO1_GDBCLK clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
18	CLKACTIVITY_GPIO_6_GDBCLK	R	1h	This field indicates the state of the GPIO6_GDBCLK clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
17	CLKACTIVITY_LCDC_GCLK	R	0h	This field indicates the state of the LCD clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
16	CLKACTIVITY_TIMER4_GCLK	R	0h	This field indicates the state of the TIMER4 CLKTIMER clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
15	CLKACTIVITY_TIMER3_GCLK	R	0h	This field indicates the state of the TIMER3 CLKTIMER clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
14	CLKACTIVITY_TIMER2_GCLK	R	0h	This field indicates the state of the TIMER2 CLKTIMER clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
13	CLKACTIVITY_TIMER7_GCLK	R	0h	This field indicates the state of the TIMER7 CLKTIMER clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
12	Reserved	R	0h	
11	CLKACTIVITY_CAN_CLK	R	0h	This field indicates the state of the CAN_CLK clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
10	CLKACTIVITY_UART_GFCLK	R	0h	This field indicates the state of the UART_GFCLK clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
9	Reserved	R	0h	
8	CLKACTIVITY_L4LS_GCLK	R	1h	This field indicates the state of the L4LS_GCLK clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
7-2	Reserved	R	0h	
1-0	CLKTRCTRL	R/W	2h	Controls the clock state transition of the L4 SLOW clock domain in PER power domain. 0 - NO_SLEEP: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1 - SW_SLEEP: SW_SLEEP: Start a software forced sleep transition on the domain. 2 - SW_WKUP: SW_WKUP: Start a software forced wake-up transition on the domain. 3 - HW_AUTO: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.2 CM_PER_L4FW_CLKCTRL Register (offset = 64h)

This register manages the L4FW clocks.

CM_PER_L4FW_CLKCTRL is shown in Figure 7-50 and described in Table 7-62.

Figure 7-50. CM_PER_L4FW_CLKCTRL Register

31	30	29	28	27	26	25	24
Reserved							
R-0h							

23	22	21	20	19	18	17	16
Reserved						IDLEST	
R-0h						R-0h	

15	14	13	12	11	10	9	8
Reserved							
R-0h							

7	6	5	4	3	2	1	0
Reserved						MODULEMODE	
R-0h						R/W-2h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-62. CM_PER_L4FW_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	0h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	2h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.3 CM_PER_MCASP1_CLKCTRL Register (offset = 68h)

This register manages the MCASP1 clocks.

CM_PER_MCASP1_CLKCTRL is shown in Figure 7-51 and described in Table 7-63.

Figure 7-51. CM_PER_MCASP1_CLKCTRL Register

31	30	29	28	27	26	25	24
Reserved							
R-0h							

23	22	21	20	19	18	17	16
Reserved						IDLEST	
R-0h						R-3h	

15	14	13	12	11	10	9	8
Reserved							
R-0h							

7	6	5	4	3	2	1	0
Reserved						MODULEMODE	
R-0h						R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-63. CM_PER_MCASP1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.4 CM_PER_UART1_CLKCTRL Register (offset = 6Ch)

This register manages the IART1 clocks.

CM_PER_UART1_CLKCTRL is shown in Figure 7-52 and described in Table 7-64.

Figure 7-52. CM_PER_UART1_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-64. CM_PER_UART1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.5 CM_PER_UART2_CLKCTRL Register (offset = 70h)

This register manages the UART2 clocks.

CM_PER_UART2_CLKCTRL is shown in Figure 7-53 and described in Table 7-65.

Figure 7-53. CM_PER_UART2_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-65. CM_PER_UART2_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.6 CM_PER_UART3_CLKCTRL Register (offset = 74h)

This register manages the UART3 clocks.

CM_PER_UART3_CLKCTRL is shown in Figure 7-54 and described in Table 7-66.

Figure 7-54. CM_PER_UART3_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-66. CM_PER_UART3_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.7 CM_PER_L3_CLKSTCTRL Register (offset = Ch)

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

CM_PER_L3_CLKSTCTRL is shown in Figure 7-55 and described in Table 7-67.

Figure 7-55. CM_PER_L3_CLKSTCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-67. CM_PER_L3_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	Reserved	R	0h	
7	CLKACTIVITY_MCASP_GCLK	R	0h	This field indicates the state of the MCASP_GCLK clock in the domain. 0 - Inact 1 - Act
6	CLKACTIVITY_CPTS_RFT_GCLK	R	0h	This field indicates the state of the CLKACTIVITY_CPTS_RFT_GCLK clock in the domain. 0 - Inact 1 - Act
5	Reserved	R	0h	
4	CLKACTIVITY_L3_GCLK	R	1h	This field indicates the state of the L3_GCLK clock in the domain. 0 - Inact 1 - Act
3	CLKACTIVITY_MMC_FCLK	R	0h	This field indicates the state of the MMC_GCLK clock in the domain. 0 - Inact 1 - Act
2	CLKACTIVITY_EMIF_GCLK	R	0h	This field indicates the state of the EMIF_GCLK clock in the domain. 0 - Inact 1 - Act
1-0	CLKTRCTRL	R/W	2h	Controls the clock state transition of the L3 clock domain. 0 - NO_SLEEP: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1 - SW_SLEEP: SW_SLEEP: Start a software forced sleep transition on the domain. 2 - SW_WKUP: SW_WKUP: Start a software forced wake-up transition on the domain. 3 - HW_AUTO: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.8 CM_PER_UART4_CLKCTRL Register (offset = 78h)

This register manages the UART4 clocks.

CM_PER_UART4_CLKCTRL is shown in Figure 7-56 and described in Table 7-68.

Figure 7-56. CM_PER_UART4_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-68. CM_PER_UART4_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.9 CM_PER_TIMER7_CLKCTRL Register (offset = 7Ch)

This register manages the TIMER7 clocks.

CM_PER_TIMER7_CLKCTRL is shown in Figure 7-57 and described in Table 7-69.

Figure 7-57. CM_PER_TIMER7_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-69. CM_PER_TIMER7_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.10 CM_PER_TIMER2_CLKCTRL Register (offset = 80h)

This register manages the TIMER2 clocks.

CM_PER_TIMER2_CLKCTRL is shown in Figure 7-58 and described in Table 7-70.

Figure 7-58. CM_PER_TIMER2_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-70. CM_PER_TIMER2_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.11 CM_PER_TIMER3_CLKCTRL Register (offset = 84h)

This register manages the TIMER3 clocks.

CM_PER_TIMER3_CLKCTRL is shown in Figure 7-59 and described in Table 7-71.

Figure 7-59. CM_PER_TIMER3_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-71. CM_PER_TIMER3_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.12 CM_PER_TIMER4_CLKCTRL Register (offset = 88h)

This register manages the TIMER4 clocks.

CM_PER_TIMER4_CLKCTRL is shown in Figure 7-60 and described in Table 7-72.

Figure 7-60. CM_PER_TIMER4_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-72. CM_PER_TIMER4_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.13 CM_PER_MCASP2_CLKCTRL Register (offset = 8Ch)

This register manages the MCASP2 clocks.

CM_PER_MCASP2_CLKCTRL is shown in Figure 7-61 and described in Table 7-73.

Figure 7-61. CM_PER_MCASP2_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-73. CM_PER_MCASP2_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.14 CM_PER_RNG_CLKCTRL Register (offset = 90h)

This register manages the RNG clocks.

CM_PER_RNG_CLKCTRL is shown in Figure 7-62 and described in Table 7-74.

Figure 7-62. CM_PER_RNG_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-74. CM_PER_RNG_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.15 CM_PER_AES0_CLKCTRL Register (offset = 94h)

This register manages the AES0 clocks.

CM_PER_AES0_CLKCTRL is shown in Figure 7-63 and described in Table 7-75.

Figure 7-63. CM_PER_AES0_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-75. CM_PER_AES0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.16 CM_PER_AES1_CLKCTRL Register (offset = 98h)

This register manages the AES1 clocks.

CM_PER_AES1_CLKCTRL is shown in Figure 7-64 and described in Table 7-76.

Figure 7-64. CM_PER_AES1_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-76. CM_PER_AES1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.17 CM_PER_DES_CLKCTRL Register (offset = 9Ch)

This register manages the DES clocks.

CM_PER_DES_CLKCTRL is shown in Figure 7-65 and described in Table 7-77.

Figure 7-65. CM_PER_DES_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-77. CM_PER_DES_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.18 Reserved Register (offset = 10h)

Reserved is shown in Figure 7-66 and described in Table 7-78.

Figure 7-66. Reserved Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-78. Reserved Register Field Descriptions

Bit	Field	Type	Reset	Description

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.19 CM_PER_SHA0_CLKCTRL Register (offset = A0h)

This register manages the SHA0 clocks.

CM_PER_SHA0_CLKCTRL is shown in Figure 7-67 and described in Table 7-79.

Figure 7-67. CM_PER_SHA0_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-79. CM_PER_SHA0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.20 CM_PER_PKA_CLKCTRL Register (offset = A4h)

This register manages the PKA clocks.

CM_PER_PKA_CLKCTRL is shown in Figure 7-68 and described in Table 7-80.

Figure 7-68. CM_PER_PKA_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-80. CM_PER_PKA_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.21 CM_PER_GPIO6_CLKCTRL Register (offset = A8h)

This register manages the GPIO6 clocks.

CM_PER_GPIO6_CLKCTRL is shown in Figure 7-69 and described in Table 7-81.

Figure 7-69. CM_PER_GPIO6_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-81. CM_PER_GPIO6_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	OPTFCLKEN_GPIO_6_GDBCLK	R/W	0h	Optional functional clock control. 0 - FCLK_DIS: Optional functional clock is disabled 1 - FCLK_EN: Optional functional clock is enabled
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.22 CM_PER_GPIO1_CLKCTRL Register (offset = ACh)

This register manages the GPIO1 clocks.

CM_PER_GPIO1_CLKCTRL is shown in Figure 7-70 and described in Table 7-82.

Figure 7-70. CM_PER_GPIO1_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-82. CM_PER_GPIO1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	OPTFCLKEN_GPIO_1_GDBCLK	R/W	0h	Optional functional clock control. 0 - FCLK_DIS: Optional functional clock is disabled 1 - FCLK_EN: Optional functional clock is enabled
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.23 CM_PER_GPIO2_CLKCTRL Register (offset = B0h)

This register manages the GPIO2 clocks.

CM_PER_GPIO2_CLKCTRL is shown in Figure 7-71 and described in Table 7-83.

Figure 7-71. CM_PER_GPIO2_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-83. CM_PER_GPIO2_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	OPTFCLKEN_GPIO_2_GDBCLK	R/W	0h	Optional functional clock control. 0 - FCLK_DIS: Optional functional clock is disabled 1 - FCLK_EN: Optional functional clock is enabled
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.24 CM_PER_GPIO3_CLKCTRL Register (offset = B4h)

This register manages the GPIO3 clocks.

CM_PER_GPIO3_CLKCTRL is shown in Figure 7-72 and described in Table 7-84.

Figure 7-72. CM_PER_GPIO3_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-84. CM_PER_GPIO3_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	OPTFCLKEN_GPIO_3_GDBCLK	R/W	0h	Optional functional clock control. 0 - FCLK_DIS: Optional functional clock is disabled 1 - FCLK_EN: Optional functional clock is enabled
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.25 CM_PER_GPIO4_CLKCTRL Register (offset = B8h)

This register manages the GPIO4 clocks.

CM_PER_GPIO4_CLKCTRL is shown in Figure 7-73 and described in Table 7-85.

Figure 7-73. CM_PER_GPIO4_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-85. CM_PER_GPIO4_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	OPTFCLKEN_GPIO_4_GDBCLK	R/W	0h	Optional functional clock control. 0 - FCLK_DIS: Optional functional clock is disabled 1 - FCLK_EN: Optional functional clock is enabled
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.26 CM_PER_TPCC_CLKCTRL Register (offset = BCh)

This register manages the TPCC clocks.

CM_PER_TPCC_CLKCTRL is shown in Figure 7-74 and described in Table 7-86.

Figure 7-74. CM_PER_TPCC_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-86. CM_PER_TPCC_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.27 CM_PER_DCAN0_CLKCTRL Register (offset = C0h)

This register manages the DCAN0 clocks.

CM_PER_DCAN0_CLKCTRL is shown in Figure 7-75 and described in Table 7-87.

Figure 7-75. CM_PER_DCAN0_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-87. CM_PER_DCAN0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.28 CM_PER_DCAN1_CLKCTRL Register (offset = C4h)

This register manages the DCAN1 clocks.

CM_PER_DCAN1_CLKCTRL is shown in Figure 7-76 and described in Table 7-88.

Figure 7-76. CM_PER_DCAN1_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-88. CM_PER_DCAN1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.29 CM_PER_CPGMAC0_CLKCTRL Register (offset = 14h)

This register manages the CPSW clocks.

CM_PER_CPGMAC0_CLKCTRL is shown in Figure 7-77 and described in Table 7-89.

Figure 7-77. CM_PER_CPGMAC0_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-89. CM_PER_CPGMAC0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	STBYST	R	1h	Module standby status. This bit is warm reset insensitive when CPSW RESET_ISO is enabled 0 - Func: Module is functional (not in standby) 1 - Standby: Module is in standby
17-16	IDLEST	R	3h	Module idle status. This bit is warm reset insensitive when CPSW RESET_ISO is enabled 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. This bit is warm reset insensitive when CPSW RESET_ISO is enabled 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.30 CM_PER_SPARE_CLKCTRL Register (offset = C8h)

This register is reserved

CM_PER_SPARE_CLKCTRL is shown in Figure 7-78 and described in Table 7-90.

Figure 7-78. CM_PER_SPARE_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-90. CM_PER_SPARE_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	STBYST	R	1h	Module standby status. 0 - Func: Module is functional (not in standby) 1 - Standby: Module is in standby
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.31 CM_PER_EPWMSS1_CLKCTRL Register (offset = CCh)

This register manages the PWMSS1 clocks.

CM_PER_EPWMSS1_CLKCTRL is shown in Figure 7-79 and described in Table 7-91.

Figure 7-79. CM_PER_EPWMSS1_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-91. CM_PER_EPWMSS1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.32 CM_PER_EMIF_FW_CLKCTRL Register (offset = D0h)

This register manages the EMIF Firewall clocks.

CM_PER_EMIF_FW_CLKCTRL is shown in Figure 7-80 and described in Table 7-92.

Figure 7-80. CM_PER_EMIF_FW_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-92. CM_PER_EMIF_FW_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.33 CM_PER_EPWMSS0_CLKCTRL Register (offset = D4h)

This register manages the PWMSS0 clocks.

CM_PER_EPWMSS0_CLKCTRL is shown in Figure 7-81 and described in Table 7-93.

Figure 7-81. CM_PER_EPWMSS0_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-93. CM_PER_EPWMSS0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.34 CM_PER_EPWMSS2_CLKCTRL Register (offset = D8h)

This register manages the PWMSS2 clocks.

CM_PER_EPWMSS2_CLKCTRL is shown in Figure 7-82 and described in Table 7-94.

Figure 7-82. CM_PER_EPWMSS2_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-94. CM_PER_EPWMSS2_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.35 CM_PER_L3_INSTR_CLKCTRL Register (offset = DCh)

This register manages the L3 INSTR clocks.

CM_PER_L3_INSTR_CLKCTRL is shown in Figure 7-83 and described in Table 7-95.

Figure 7-83. CM_PER_L3_INSTR_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-95. CM_PER_L3_INSTR_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	0h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	2h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.36 CM_PER_L3_CLKCTRL Register (offset = E0h)

This register manages the L3 Interconnect clocks.

CM_PER_L3_CLKCTRL is shown in Figure 7-84 and described in Table 7-96.

Figure 7-84. CM_PER_L3_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-96. CM_PER_L3_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	0h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	2h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.37 CM_PER_IEEE5000_CLKCTRL Register (offset = E4h)

This register manages the IEEE1500 clocks.

CM_PER_IEEE5000_CLKCTRL is shown in Figure 7-85 and described in Table 7-97.

Figure 7-85. CM_PER_IEEE5000_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-97. CM_PER_IEEE5000_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	STBYST	R	1h	Module standby status. 0 - Func: Module is functional (not in standby) 1 - Standby: Module is in standby
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	2h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.38 CM_PER_ICSS_CLKCTRL Register (offset = E8h)

This register manages the ICSS clocks.

CM_PER_ICSS_CLKCTRL is shown in Figure 7-86 and described in Table 7-98.

Figure 7-86. CM_PER_ICSS_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-98. CM_PER_ICSS_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	STBYST	R	1h	Module standby status. 0 - Func: Module is functional (not in standby) 1 - Standby: Module is in standby
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.39 CM_PER_TIMER5_CLKCTRL Register (offset = ECh)

This register manages the TIMER5 clocks.

CM_PER_TIMER5_CLKCTRL is shown in Figure 7-87 and described in Table 7-99.

Figure 7-87. CM_PER_TIMER5_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-99. CM_PER_TIMER5_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.40 CM_PER_LCDC_CLKCTRL Register (offset = 18h)

This register manages the LCD clocks.

CM_PER_LCDC_CLKCTRL is shown in Figure 7-88 and described in Table 7-100.

Figure 7-88. CM_PER_LCDC_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-100. CM_PER_LCDC_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	STBYST	R	1h	Module standby status. 0 - Func: Module is functional (not in standby) 1 - Standby: Module is in standby
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.41 CM_PER_TIMER6_CLKCTRL Register (offset = F0h)

This register manages the TIMER6 clocks.

CM_PER_TIMER6_CLKCTRL is shown in Figure 7-89 and described in Table 7-101.

Figure 7-89. CM_PER_TIMER6_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-101. CM_PER_TIMER6_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.42 CM_PER_MMC1_CLKCTRL Register (offset = F4h)

This register manages the MMC1 clocks.

CM_PER_MMC1_CLKCTRL is shown in Figure 7-90 and described in Table 7-102.

Figure 7-90. CM_PER_MMC1_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-102. CM_PER_MMC1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.43 CM_PER_MMC2_CLKCTRL Register (offset = F8h)

This register manages the MMC2 clocks.

CM_PER_MMC2_CLKCTRL is shown in Figure 7-91 and described in Table 7-103.

Figure 7-91. CM_PER_MMC2_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-103. CM_PER_MMC2_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.44 CM_PER_TPTC1_CLKCTRL Register (offset = FCh)

This register manages the TPTC1 clocks.

CM_PER_TPTC1_CLKCTRL is shown in Figure 7-92 and described in Table 7-104.

Figure 7-92. CM_PER_TPTC1_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-104. CM_PER_TPTC1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	STBYST	R	1h	Module standby status. 0 - Func: Module is functional (not in standby) 1 - Standby: Module is in standby
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.45 CM_PER_TPTC2_CLKCTRL Register (offset = 100h)

This register manages the TPTC2 clocks.

CM_PER_TPTC2_CLKCTRL is shown in Figure 7-93 and described in Table 7-105.

Figure 7-93. CM_PER_TPTC2_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-105. CM_PER_TPTC2_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	STBYST	R	1h	Module standby status. 0 - Func: Module is functional (not in standby) 1 - Standby: Module is in standby
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.46 CM_PER_GPIO5_CLKCTRL Register (offset = 104h)

This register manages the GPIO5 clocks.

CM_PER_GPIO5_CLKCTRL is shown in Figure 7-94 and described in Table 7-106.

Figure 7-94. CM_PER_GPIO5_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-106. CM_PER_GPIO5_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	OPTFCLKEN_GPIO_5_GDBCLK	R/W	0h	Optional functional clock control. 0 - FCLK_DIS: Optional functional clock is disabled 1 - FCLK_EN: Optional functional clock is enabled
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.47 CM_PER_SPINLOCK_CLKCTRL Register (offset = 10Ch)

This register manages the SPINLOCK clocks.

CM_PER_SPINLOCK_CLKCTRL is shown in Figure 7-95 and described in Table 7-107.

Figure 7-95. CM_PER_SPINLOCK_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-107. CM_PER_SPINLOCK_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.48 CM_PER_MAILBOX0_CLKCTRL Register (offset = 110h)

This register manages the MAILBOX0 clocks.

CM_PER_MAILBOX0_CLKCTRL is shown in Figure 7-96 and described in Table 7-108.

Figure 7-96. CM_PER_MAILBOX0_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-108. CM_PER_MAILBOX0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.49 CM_PER_USB0_CLKCTRL Register (offset = 1Ch)

This register manages the USB clocks.

CM_PER_USB0_CLKCTRL is shown in Figure 7-97 and described in Table 7-109.

Figure 7-97. CM_PER_USB0_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-109. CM_PER_USB0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	STBYST	R	1h	Module standby status. 0 - Func: Module is functional (not in standby) 1 - Standby: Module is in standby
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.50 CM_PER_L4HS_CLKSTCTRL Register (offset = 11Ch)

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

CM_PER_L4HS_CLKSTCTRL is shown in Figure 7-98 and described in Table 7-110.

Figure 7-98. CM_PER_L4HS_CLKSTCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-110. CM_PER_L4HS_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	Reserved	R	0h	
6	CLKACTIVITY_CPSW_5MHZ_GCLK	R	1h	This field indicates the state of the CPSW_5MHZ_GCLK clock in the domain. 0 - Inact 1 - Act
5	CLKACTIVITY_CPSW_50MHZ_GCLK	R	1h	This field indicates the state of the CPSW_50MHZ_GCLK clock in the domain. 0 - Inact 1 - Act
4	CLKACTIVITY_CPSW_250MHZ_GCLK	R	1h	This field indicates the state of the CPSW_250MHZ_GCLK clock in the domain. 0 - Inact 1 - Act
3	CLKACTIVITY_L4HS_GCLK	R	1h	This field indicates the state of the L4HS_GCLK clock in the domain. 0 - Inact 1 - Act
2	Reserved	R	0h	
1-0	CLKTRCTRL	R/W	2h	Controls the clock state transition of the L4 Fast clock domain. 0 - NO_SLEEP: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1 - SW_SLEEP: SW_SLEEP: Start a software forced sleep transition on the domain. 2 - SW_WKUP: SW_WKUP: Start a software forced wake-up transition on the domain. 3 - HW_AUTO: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.51 CM_PER_L4HS_CLKCTRL Register (offset = 120h)

This register manages the L4 Fast clocks.

CM_PER_L4HS_CLKCTRL is shown in Figure 7-99 and described in Table 7-111.

Figure 7-99. CM_PER_L4HS_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-111. CM_PER_L4HS_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	0h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	2h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.52 CM_PER_MSTR_EXPS_CLKCTRL Register (offset = 124h)

This register manages the Master Expansion Port clocks.

CM_PER_MSTR_EXPS_CLKCTRL is shown in Figure 7-100 and described in Table 7-112.

Figure 7-100. CM_PER_MSTR_EXPS_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-112. CM_PER_MSTR_EXPS_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	STBYST	R	1h	Module standby status. 0 - Func: Module is functional (not in standby) 1 - Standby: Module is in standby
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.53 CM_PER_SLV_EXPS_CLKCTRL Register (offset = 128h)

This register manages the Slave Expansion Port clocks.

CM_PER_SLV_EXPS_CLKCTRL is shown in Figure 7-101 and described in Table 7-113.

Figure 7-101. CM_PER_SLV_EXPS_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-113. CM_PER_SLV_EXPS_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	0h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.54 CM_PER_OCPWP_L3_CLKSTCTRL Register (offset = 12Ch)

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

CM_PER_OCPWP_L3_CLKSTCTRL is shown in Figure 7-102 and described in Table 7-114.

Figure 7-102. CM_PER_OCPWP_L3_CLKSTCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-114. CM_PER_OCPWP_L3_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	Reserved	R	0h	
5	CLKACTIVITY_OCPWP_L4_GCLK	R	0h	This field indicates the state of the OCPWP L4 clock in the domain. 0 - Inact 1 - Act
4	CLKACTIVITY_OCPWP_L3_GCLK	R	0h	This field indicates the state of the OCPWP L3 clock in the domain. 0 - Inact 1 - Act
3-2	Reserved	R	0h	
1-0	CLKTRCTRL	R/W	2h	Controls the clock state transition of the OCPWP clock domain. 0 - NO_SLEEP: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1 - SW_SLEEP: SW_SLEEP: Start a software forced sleep transition on the domain. 2 - SW_WKUP: SW_WKUP: Start a software forced wake-up transition on the domain. 3 - HW_AUTO: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.55 CM_PER_OCPWP_CLKCTRL Register (offset = 130h)

This register manages the OCPWP clocks.

CM_PER_OCPWP_CLKCTRL is shown in Figure 7-103 and described in Table 7-115.

Figure 7-103. CM_PER_OCPWP_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-115. CM_PER_OCPWP_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	STBYST	R	1h	Module standby status. 0 - Func: Module is functional (not in standby) 1 - Standby: Module is in standby
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	2h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.56 CM_PER_MAILBOX1_CLKCTRL Register (offset = 134h)

This register manages the Mailbox1 clocks.

CM_PER_MAILBOX1_CLKCTRL is shown in Figure 7-104 and described in Table 7-116.

Figure 7-104. CM_PER_MAILBOX1_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-116. CM_PER_MAILBOX1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.57 CM_PER_SPARE0_CLKCTRL Register (offset = 138h)

This register manages the Spare0 clocks.

CM_PER_SPARE0_CLKCTRL is shown in Figure 7-105 and described in Table 7-117.

Figure 7-105. CM_PER_SPARE0_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-117. CM_PER_SPARE0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.58 CM_PER_SPARE1_CLKCTRL Register (offset = 13Ch)

This register manages the Spare1 clocks.

CM_PER_SPARE1_CLKCTRL is shown in Figure 7-106 and described in Table 7-118.

Figure 7-106. CM_PER_SPARE1_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-118. CM_PER_SPARE1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.59 CM_PER_MLB_CLKCTRL Register (offset = 20h)

This register manages the MLB clocks.

CM_PER_MLB_CLKCTRL is shown in Figure 7-107 and described in Table 7-119.

Figure 7-107. CM_PER_MLB_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-119. CM_PER_MLB_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	STBYST	R	1h	Module standby status. 0 - Func: Module is functional (not in standby) 1 - Standby: Module is in standby
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.60 CM_PER_ICSS_CLKSTCTRL Register (offset = 140h)

This register enables the clock domain state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

CM_PER_ICSS_CLKSTCTRL is shown in Figure 7-108 and described in Table 7-120.

Figure 7-108. CM_PER_ICSS_CLKSTCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-120. CM_PER_ICSS_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	Reserved	R	0h	
6	CLKACTIVITY_ICSS_UART_GCLK	R	0h	This field indicates the state of the ICSS UART clock in the domain. 0 - Inact 1 - Act
5	CLKACTIVITY_ICSS_IEP_GCLK	R	0h	This field indicates the state of the ICSS IEP clock in the domain. 0 - Inact 1 - Act
4	CLKACTIVITY_ICSS_OCP_GCLK	R	0h	This field indicates the state of the ICSS OCP clock in the domain. 0 - Inact 1 - Act
3-2	Reserved	R	0h	
1-0	CLKTRCTRL	R/W	2h	Controls the clock state transition of the ICSS OCP clock domain. 0 - NO_SLEEP: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1 - SW_SLEEP: SW_SLEEP: Start a software forced sleep transition on the domain. 2 - SW_WKUP: SW_WKUP: Start a software forced wake-up transition on the domain. 3 - HW_AUTO: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.61 CM_PER_CPSW_CLKSTCTRL Register (offset = 144h)

This register enables the clock domain state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

CM_PER_CPSW_CLKSTCTRL is shown in Figure 7-109 and described in Table 7-121.

Figure 7-109. CM_PER_CPSW_CLKSTCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-121. CM_PER_CPSW_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	Reserved	R	0h	
4	CLKACTIVITY_CPSW_125MHz_GCLK	R	0h	This field indicates the state of the CPSW 125 MHz OCP clock in the domain. 0 - Inact 1 - Act
3-2	Reserved	R	0h	
1-0	CLKTRCTRL	R/W	2h	Controls the clock state transition of the CPSW OCP clock domain. 0 - NO_SLEEP: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1 - SW_SLEEP: SW_SLEEP: Start a software forced sleep transition on the domain. 2 - SW_WKUP: SW_WKUP: Start a software forced wake-up transition on the domain. 3 - HW_AUTO: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.62 CM_PER_LCDC_CLKSTCTRL Register (offset = 148h)

This register enables the clock domain state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

CM_PER_LCDC_CLKSTCTRL is shown in Figure 7-110 and described in Table 7-122.

Figure 7-110. CM_PER_LCDC_CLKSTCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-122. CM_PER_LCDC_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	Reserved	R	0h	
5	CLKACTIVITY_LCDC_L4_OCP_GCLK	R	0h	This field indicates the state of the LCDC L4 OCP clock in the domain. 0 - Inact 1 - Act
4	CLKACTIVITY_LCDC_L3_OCP_GCLK	R	0h	This field indicates the state of the LCDC L3 OCP clock in the domain. 0 - Inact 1 - Act
3-2	Reserved	R	0h	
1-0	CLKTRCTRL	R/W	2h	Controls the clock state transition of the LCDC OCP clock domain. 0 - NO_SLEEP: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1 - SW_SLEEP: SW_SLEEP: Start a software forced sleep transition on the domain. 2 - SW_WKUP: SW_WKUP: Start a software forced wake-up transition on the domain. 3 - HW_AUTO: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.63 CM_PER_CLKDIV32K_CLKCTRL Register (offset = 14Ch)

This register manages the CLKDIV32K clocks.

CM_PER_CLKDIV32K_CLKCTRL is shown in Figure 7-111 and described in Table 7-123.

Figure 7-111. CM_PER_CLKDIV32K_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-123. CM_PER_CLKDIV32K_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.64 CM_PER_CLK_24MHZ_CLKSTCTRL Register (offset = 150h)

This register enables the clock domain state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

CM_PER_CLK_24MHZ_CLKSTCTRL is shown in Figure 7-112 and described in Table 7-124.

Figure 7-112. CM_PER_CLK_24MHZ_CLKSTCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-124. CM_PER_CLK_24MHZ_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-5	Reserved	R	0h	
4	CLKACTIVITY_CLK_24MHZ_GCLK	R	0h	This field indicates the state of the 24MHz clock in the domain. 0 - Inact 1 - Act
3-2	Reserved	R	0h	
1-0	CLKTRCTRL	R/W	2h	Controls the clock state transition of the 24MHz clock domain. 0 - NO_SLEEP: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1 - SW_SLEEP: SW_SLEEP: Start a software forced sleep transition on the domain. 2 - SW_WKUP: SW_WKUP: Start a software forced wake-up transition on the domain. 3 - HW_AUTO: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.65 CM_PER_TPTC0_CLKCTRL Register (offset = 24h)

This register manages the TPTC clocks.

CM_PER_TPTC0_CLKCTRL is shown in Figure 7-113 and described in Table 7-125.

Figure 7-113. CM_PER_TPTC0_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-125. CM_PER_TPTC0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	STBYST	R	1h	Module standby status. 0 - Func: Module is functional (not in standby) 1 - Standby: Module is in standby
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.66 CM_PER_L3S_CLKSTCTRL Register (offset = 4h)

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

CM_PER_L3S_CLKSTCTRL is shown in Figure 7-114 and described in Table 7-126.

Figure 7-114. CM_PER_L3S_CLKSTCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-126. CM_PER_L3S_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	R	XXXXh	
25-11	Reserved	R	0h	
10	Reserved	R	0h	
9	Reserved	R	0h	
8	Reserved	R	0h	
7-5	Reserved	R	0h	
4	Reserved	R	0h	
3	CLKACTIVITY_L3S_GCLK	R	1h	This field indicates the state of the L3S_GCLK clock in the domain. 0 - Inact 1 - Act
2	Reserved	R	0h	
1-0	CLKTRCTRL	R/W	2h	Controls the clock state transition of the L3 Slow clock domain. 0 - NO_SLEEP: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1 - SW_SLEEP: SW_SLEEP: Start a software forced sleep transition on the domain. 2 - SW_WKUP: SW_WKUP: Start a software forced wake-up transition on the domain. 3 - HW_AUTO: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.67 CM_PER_EMIF_CLKCTRL Register (offset = 28h)

This register manages the EMIF clocks.

CM_PER_EMIF_CLKCTRL is shown in Figure 7-115 and described in Table 7-127.

Figure 7-115. CM_PER_EMIF_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-127. CM_PER_EMIF_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.68 CM_PER_OCMCRAM_CLKCTRL Register (offset = 2Ch)

This register manages the OCMC clocks.

CM_PER_OCMCRAM_CLKCTRL is shown in Figure 7-116 and described in Table 7-128.

Figure 7-116. CM_PER_OCMCRAM_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-128. CM_PER_OCMCRAM_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.69 CM_PER_GPMC_CLKCTRL Register (offset = 30h)

This register manages the GPMC clocks.

CM_PER_GPMC_CLKCTRL is shown in Figure 7-117 and described in Table 7-129.

Figure 7-117. CM_PER_GPMC_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-129. CM_PER_GPMC_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	2h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.70 CM_PER_MCASP0_CLKCTRL Register (offset = 34h)

This register manages the MCASP0 clocks.

CM_PER_MCASP0_CLKCTRL is shown in Figure 7-118 and described in Table 7-130.

Figure 7-118. CM_PER_MCASP0_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-130. CM_PER_MCASP0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.71 CM_PER_UART5_CLKCTRL Register (offset = 38h)

This register manages the UART5 clocks.

CM_PER_UART5_CLKCTRL is shown in Figure 7-119 and described in Table 7-131.

Figure 7-119. CM_PER_UART5_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-131. CM_PER_UART5_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.72 CM_PER_MMC0_CLKCTRL Register (offset = 3Ch)

This register manages the MMC0 clocks.

CM_PER_MMC0_CLKCTRL is shown in Figure 7-120 and described in Table 7-132.

Figure 7-120. CM_PER_MMC0_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-132. CM_PER_MMC0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.73 CM_PER_ELM_CLKCTRL Register (offset = 40h)

This register manages the ELM clocks.

CM_PER_ELM_CLKCTRL is shown in Figure 7-121 and described in Table 7-133.

Figure 7-121. CM_PER_ELM_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-133. CM_PER_ELM_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.74 CM_PER_I2C2_CLKCTRL Register (offset = 44h)

This register manages the I2C2 clocks.

CM_PER_I2C2_CLKCTRL is shown in Figure 7-122 and described in Table 7-134.

Figure 7-122. CM_PER_I2C2_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-134. CM_PER_I2C2_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.75 CM_PER_I2C1_CLKCTRL Register (offset = 48h)

This register manages the I2C1 clocks.

CM_PER_I2C1_CLKCTRL is shown in Figure 7-123 and described in Table 7-135.

Figure 7-123. CM_PER_I2C1_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-135. CM_PER_I2C1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.76 CM_PER_SPI0_CLKCTRL Register (offset = 4Ch)

This register manages the SPI0 clocks.

CM_PER_SPI0_CLKCTRL is shown in Figure 7-124 and described in Table 7-136.

Figure 7-124. CM_PER_SPI0_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-136. CM_PER_SPI0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.77 CM_PER_L4FW_CLKSTCTRL Register (offset = 8h)

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

CM_PER_L4FW_CLKSTCTRL is shown in Figure 7-125 and described in Table 7-137.

Figure 7-125. CM_PER_L4FW_CLKSTCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-137. CM_PER_L4FW_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	R	0h	
25-11	Reserved	R	0h	
10	Reserved	R	0h	
9	Reserved	R	0h	
8	CLKACTIVITY_L4FW_GCLK	R	1h	This field indicates the state of the L4FW clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
7-2	Reserved	R	0h	
1-0	CLKTRCTRL	R/W	2h	Controls the clock state transition of the L4 Firewall clock domain. 0 - NO_SLEEP: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1 - SW_SLEEP: SW_SLEEP: Start a software forced sleep transition on the domain. 2 - SW_WKUP: SW_WKUP: Start a software forced wake-up transition on the domain. 3 - HW_AUTO: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.78 CM_PER_SPI1_CLKCTRL Register (offset = 50h)

This register manages the SPI1 clocks.

CM_PER_SPI1_CLKCTRL is shown in Figure 7-126 and described in Table 7-138.

Figure 7-126. CM_PER_SPI1_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-138. CM_PER_SPI1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.79 CM_PER_SPI2_CLKCTRL Register (offset = 54h)

This register manages the SPI2 clocks.

CM_PER_SPI2_CLKCTRL is shown in Figure 7-127 and described in Table 7-139.

Figure 7-127. CM_PER_SPI2_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-139. CM_PER_SPI2_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.80 CM_PER_SPI3_CLKCTRL Register (offset = 58h)

This register manages the SPI3 clocks.

CM_PER_SPI3_CLKCTRL is shown in Figure 7-128 and described in Table 7-140.

Figure 7-128. CM_PER_SPI3_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-140. CM_PER_SPI3_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.10.81 CM_PER_L4LS_CLKCTRL Register (offset = 60h)

This register manages the L4LS clocks.

CM_PER_L4LS_CLKCTRL is shown in Figure 7-129 and described in Table 7-141.

Figure 7-129. CM_PER_L4LS_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-141. CM_PER_L4LS_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	0h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	2h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

7.2.5.0.11 CM_RTC Registers

Table 7-142 lists the memory-mapped registers for the CM_RTC. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 7-142 should be considered as reserved locations and the register contents should not be modified.

Table 7-142. CM_RTC REGISTERS

Offset	Acronym	Register Description	Section
0h	CM_RTC_RTC_CLKCTRL	This register manages the RTC clocks.	Section 7.2.5.0.11.1
4h	CM_RTC_CLKSTCTRL	This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.	Section 7.2.5.0.11.2

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.11.1 CM_RTC_RTC_CLKCTRL Register (offset = 0h) [reset = 0h]

This register manages the RTC clocks.

CM_RTC_RTC_CLKCTRL is shown in Figure 7-130 and described in Table 7-143.

Figure 7-130. CM_RTC_RTC_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-143. CM_RTC_RTC_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	XXXh	
18	Reserved	R	XXXh	
17-16	IDLEST	R	XXXh	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	XXXh	
1-0	MODULEMODE	R/W	XXXh	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.11.2 CM_RTC_CLKSTCTRL Register (offset = 4h) [reset = 0h]

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

CM_RTC_CLKSTCTRL is shown in Figure 7-131 and described in Table 7-144.

Figure 7-131. CM_RTC_CLKSTCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-144. CM_RTC_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	R	XXXh	
25-11	Reserved	R	XXXh	
10	Reserved	R	XXXh	
9	CLKACTIVITY_RTC_32K CLK	R	XXXh	This field indicates the state of the 32K RTC clock in the domain. 0 - Inact 1 - Act
8	CLKACTIVITY_L4_RTC_ GCLK	R	XXXh	This field indicates the state of the L4 RTC clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
7-2	Reserved	R	XXXh	
1-0	CLKTRCTRL	R/W	XXXh	Controls the clock state transition of the RTC clock domains. 0 - NO_SLEEP: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1 - SW_SLEEP: SW_SLEEP: Start a software forced sleep transition on the domain. 2 - SW_WKUP: SW_WKUP: Start a software forced wake-up transition on the domain. 3 - HW_AUTO: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12 CM_WKUP Registers

Table 7-145 lists the memory-mapped registers for the CM_WKUP. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 7-145 should be considered as reserved locations and the register contents should not be modified.

Table 7-145. CM_WKUP REGISTERS

Offset	Acronym	Register Description	Section
0h	CM_WKUP_CLKSTCTRL	This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.	Section 7.2.5.0.12.1
64h	CM_SSC_MODFREQDIV_DPLL_CORE	Control the Modulation Frequency (Fm) for Spread Spectrum Clocking technique by defining it as a ratio of DPLL_REFCLK/4 Fm = [DPLL_REFCLK/4]/MODFREQDIV MODFREQDIV = MODFREQDIV_MANTISSA * 2^MODFREQDIV_EXPONENT [warm reset insensitive]	Section 7.2.5.0.12.2
68h	CM_CLKSEL_DPLL_CORE	This register provides controls over the DPLL.	Section 7.2.5.0.12.3
6Ch	CM_AUTOIDLE_DPLL_PER	This register provides automatic control over the DPLL activity.	Section 7.2.5.0.12.4
70h	CM_IDLEST_DPLL_PER	This register allows monitoring the master clock activity. This register is read only and automatically updated. [warm reset insensitive]	Section 7.2.5.0.12.5
74h	CM_SSC_DELTAMSTEP_DPLL_PER	Control the DeltaMStep parameter for Spread Spectrum Clocking technique DeltaMStep is split into fractional and integer part. [warm reset insensitive]	Section 7.2.5.0.12.6
Ch	CM_WKUP_L4WKUP_CLKCTRL	This register manages the L4WKUP clocks.	Section 7.2.5.0.12.7
78h	CM_SSC_MODFREQDIV_DPLL_PER	Control the Modulation Frequency (Fm) for Spread Spectrum Clocking technique by defining it as a ratio of DPLL_REFCLK/4 Fm = [DPLL_REFCLK/4]/MODFREQDIV MODFREQDIV = MODFREQDIV_MANTISSA * 2^MODFREQDIV_EXPONENT [warm reset insensitive]	Section 7.2.5.0.12.8
7Ch	CM_CLKDCOLDO_DPLL_PER	This register provides controls over the M2 divider of the DPLL.	Section 7.2.5.0.12.9
80h	CM_DIV_M4_DPLL_CORE	This register provides controls over the CLKOUT1 o/p of the HSDIVIDER.	Section 7.2.5.0.12.10
84h	CM_DIV_M5_DPLL_CORE	This register provides controls over the CLKOUT2 o/p of the HSDIVIDER.	Section 7.2.5.0.12.11
88h	CM_CLKMODE_DPLL_MPU	This register allows controlling the DPLL modes.	Section 7.2.5.0.12.12
8Ch	CM_CLKMODE_DPLL_PER	This register allows controlling the DPLL modes.	Section 7.2.5.0.12.13
90h	CM_CLKMODE_DPLL_CORE	This register allows controlling the DPLL modes.	Section 7.2.5.0.12.14
94h	CM_CLKMODE_DPLL_DDR	This register allows controlling the DPLL modes.	Section 7.2.5.0.12.15
98h	CM_CLKMODE_DPLL_DISP	This register allows controlling the DPLL modes.	Section 7.2.5.0.12.16
9Ch	CM_CLKSEL_DPLL_PERIPH	This register provides controls over the DPLL.	Section 7.2.5.0.12.17
10h	CM_WKUP_TIMER0_CLKCTRL	This register manages the TIMER0 clocks.	Section 7.2.5.0.12.18
A0h	CM_DIV_M2_DPLL_DDR	This register provides controls over the M2 divider of the DPLL.	Section 7.2.5.0.12.19
A4h	CM_DIV_M2_DPLL_DISP	This register provides controls over the M2 divider of the DPLL.	Section 7.2.5.0.12.20
A8h	CM_DIV_M2_DPLL_MPU	This register provides controls over the M2 divider of the DPLL.	Section 7.2.5.0.12.21

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 7-145. CM_WKUP REGISTERS (continued)

Offset	Acronym	Register Description	Section
ACh	CM_DIV_M2_DPLL_PER	This register provides controls over the M2 divider of the DPLL.	Section 7.2.5.0.12.22
B0h	CM_WKUP_WKUP_M3_CLKCTRL	This register manages the WKUP M3 clocks.	Section 7.2.5.0.12.23
B4h	CM_WKUP_UART0_CLKCTRL	This register manages the UART0 clocks.	Section 7.2.5.0.12.24
B8h	CM_WKUP_I2C0_CLKCTRL	This register manages the I2C0 clocks.	Section 7.2.5.0.12.25
BCh	CM_WKUP_ADC_TSC_CLKCTRL	This register manages the ADC clocks.	Section 7.2.5.0.12.26
C0h	CM_WKUP_SMARTREFLEX0_CLKCTRL	This register manages the SmartReflex0 clocks.	Section 7.2.5.0.12.27
C4h	CM_WKUP_TIMER1_CLKCTRL	This register manages the TIMER1 clocks.	Section 7.2.5.0.12.28
14h	CM_WKUP_DEBUGSS_CLKCTRL	This register manages the DEBUGSS clocks.	Section 7.2.5.0.12.29
C8h	CM_WKUP_SMARTREFLEX1_CLKCTRL	This register manages the SmartReflex1 clocks.	Section 7.2.5.0.12.30
CCh	CM_L4_WKUP_AON_CLKSTCTRL	This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.	Section 7.2.5.0.12.31
D0h	CM_WKUP_WDT0_CLKCTRL	This register manages the WDT0 clocks.	Section 7.2.5.0.12.32
D4h	CM_WKUP_WDT1_CLKCTRL	This register manages the WDT1 clocks.	Section 7.2.5.0.12.33
D8h	CM_DIV_M6_DPLL_CORE	This register provides controls over the CLKOUT3 o/p of the HSDIVIDER. [warm reset insensitive]	Section 7.2.5.0.12.34
18h	CM_L3_AON_CLKSTCTRL	This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.	Section 7.2.5.0.12.35
1Ch	CM_AUTOIDLE_DPLL_MPU	This register provides automatic control over the DPLL activity.	Section 7.2.5.0.12.36
20h	CM_IDLEST_DPLL_MPU	This register allows monitoring the master clock activity. This register is read only and automatically updated.[warm reset insensitive]	Section 7.2.5.0.12.37
24h	CM_SSC_DELTAMSTEP_DPLL_MPU	Control the DeltaMStep parameter for Spread Spectrum Clocking technique DeltaMStep is split into fractional and integer part. [warm reset insensitive]	Section 7.2.5.0.12.38
4h	CM_WKUP_CONTROL_CLKCTRL	This register manages the Control Module clocks.	Section 7.2.5.0.12.39
28h	CM_SSC_MODFREQDIV_DPLL_MPU	Control the Modulation Frequency (Fm) for Spread Spectrum Clocking technique by defining it as a ratio of DPLL_REFCLK/4 Fm = [DPLL_REFCLK/4]/MODFREQDIV MODFREQDIV = MODFREQDIV_MANTISSA * 2^MODFREQDIV_EXPONENT [warm reset insensitive]	Section 7.2.5.0.12.40
2Ch	CM_CLKSEL_DPLL_MPU	This register provides controls over the DPLL.	Section 7.2.5.0.12.41
30h	CM_AUTOIDLE_DPLL_DDR	This register provides automatic control over the DPLL activity.	Section 7.2.5.0.12.42
34h	CM_IDLEST_DPLL_DDR	This register allows monitoring the master clock activity. This register is read only and automatically updated. [warm reset insensitive]	Section 7.2.5.0.12.43

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 7-145. CM_WKUP REGISTERS (continued)

Offset	Acronym	Register Description	Section
38h	CM_SSC_DELTAMSTEP_DPLL_DDR	Control the DeltaMStep parameter for Spread Spectrum Clocking technique DeltaMStep is split into fractional and integer part. [warm reset insensitive]	Section 7.2.5.0.12.44
3Ch	CM_SSC_MODFREQDIV_DPLL_DDR	Control the Modulation Frequency (Fm) for Spread Spectrum Clocking technique by defining it as a ratio of DPLL_REFCLK/4 Fm = [DPLL_REFCLK/4]/MODFREQDIV MODFREQDIV = MODFREQDIV_MANTISSA * 2^MODFREQDIV_EXPONENT [warm reset insensitive]	Section 7.2.5.0.12.45
40h	CM_CLKSEL_DPLL_DDR	This register provides controls over the DPLL.	Section 7.2.5.0.12.46
44h	CM_AUTOIDLE_DPLL_DISP	This register provides automatic control over the DPLL activity.	Section 7.2.5.0.12.47
48h	CM_IDLEST_DPLL_DISP	This register allows monitoring the master clock activity. This register is read only and automatically updated. [warm reset insensitive]	Section 7.2.5.0.12.48
4Ch	CM_SSC_DELTAMSTEP_DPLL_DISP	Control the DeltaMStep parameter for Spread Spectrum Clocking technique DeltaMStep is split into fractional and integer part. [warm reset insensitive]	Section 7.2.5.0.12.49
8h	CM_WKUP_GPIO0_CLKCTRL	This register manages the GPIO0 clocks.	Section 7.2.5.0.12.50
50h	CM_SSC_MODFREQDIV_DPLL_DISP	Control the Modulation Frequency (Fm) for Spread Spectrum Clocking technique by defining it as a ratio of DPLL_REFCLK/4 Fm = [DPLL_REFCLK/4]/MODFREQDIV MODFREQDIV = MODFREQDIV_MANTISSA * 2^MODFREQDIV_EXPONENT [warm reset insensitive]	Section 7.2.5.0.12.51
54h	CM_CLKSEL_DPLL_DISP	This register provides controls over the DPLL.	Section 7.2.5.0.12.52
58h	CM_AUTOIDLE_DPLL_CORE	This register provides automatic control over the DPLL activity.	Section 7.2.5.0.12.53
5Ch	CM_IDLEST_DPLL_CORE	This register allows monitoring the master clock activity. This register is read only and automatically updated. [warm reset insensitive]	Section 7.2.5.0.12.54
60h	CM_SSC_DELTAMSTEP_DPLL_CORE	Control the DeltaMStep parameter for Spread Spectrum Clocking technique DeltaMStep is split into fractional and integer part. [warm reset insensitive]	Section 7.2.5.0.12.55

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.1 CM_WKUP_CLKSTCTRL Register (offset = 0h)

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

CM_WKUP_CLKSTCTRL is shown in Figure 7-132 and described in Table 7-146.

Figure 7-132. CM_WKUP_CLKSTCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-146. CM_WKUP_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	Reserved	R	0h	
14	CLKACTIVITY_ADC_FCLK	R	0h	This field indicates the state of the ADC clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
13	CLKACTIVITY_TIMER1_GCLK	R	0h	This field indicates the state of the TIMER1 clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
12	CLKACTIVITY_UART0_GFCLK	R	0h	This field indicates the state of the UART0 clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
11	CLKACTIVITY_I2C0_GFCLK	R	0h	This field indicates the state of the I2C0 clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
10	CLKACTIVITY_TIMER0_GCLK	R	0h	This field indicates the state of the WKUPTIMER_GCLK clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
9	CLKACTIVITY_WDT0_GCLK	R	0h	This field indicates the state of the WDT0_GCLK clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
8	CLKACTIVITY_GPIO0_GDBCLK	R	0h	This field indicates the state of the WKUPGPIO_DBGICLK clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
7-5	Reserved	R	0h	
4	CLKACTIVITY_WDT1_GCLK	R	0h	This field indicates the state of the WDT1_GCLK clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
3	CLKACTIVITY_SR_SYSCLK	R	0h	This field indicates the state of the SMARTREFGLEX SYSCLK clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
2	CLKACTIVITY_L4_WKUP_GCLK	R	1h	This field indicates the state of the L4_WKUP clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
1-0	CLKTRCTRL	R/W	2h	Controls the clock state transition of the always on clock domain. 0 - NO_SLEEP: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1 - SW_SLEEP: SW_SLEEP: Start a software forced sleep transition on the domain. 2 - SW_WKUP: SW_WKUP: Start a software forced wake-up transition on the domain. 3 - HW_AUTO: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.2 CM_SSC_MODFREQDIV_DPLL_CORE Register (offset = 64h)

Control the Modulation Frequency (Fm) for Spread Spectrum Clocking technique by defining it as a ratio of DPLL_REFCLK/4 Fm = [DPLL_REFCLK/4]/MODFREQDIV MODFREQDIV = MODFREQDIV_MANTISSA * 2^MODFREQDIV_EXPONENT [warm reset insensitive]

CM_SSC_MODFREQDIV_DPLL_CORE is shown in Figure 7-133 and described in Table 7-147.

Figure 7-133. CM_SSC_MODFREQDIV_DPLL_CORE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-147. CM_SSC_MODFREQDIV_DPLL_CORE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	Reserved	R	0h	
10-8	MODFREQDIV_EXPONENT	R/W	0h	Set the Exponent component of MODFREQDIV factor
7	Reserved	R	0h	
6-0	MODFREQDIV_MANTISSA	R/W	0h	Set the Mantissa component of MODFREQDIV factor

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.3 CM_CLKSEL_DPLL_CORE Register (offset = 68h)

This register provides controls over the DPLL.

CM_CLKSEL_DPLL_CORE is shown in Figure 7-134 and described in Table 7-148.

Figure 7-134. CM_CLKSEL_DPLL_CORE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-148. CM_CLKSEL_DPLL_CORE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-23	Reserved	R	0h	
22-19	Reserved	R	0h	
18-8	DPLL_MULT	R/W	0h	DPLL multiplier factor (2 to 2047). This register is automatically cleared to 0 when the DPLL_EN field in the *CLKMODE_DPLL* register is set to select MN Bypass mode. (equal to input M of DPLL; M=2 to 2047 => DPLL multiplies by M) 0 - Reserved_0: Reserved 1 - Reserved_1: Reserved
7	Reserved	R	0h	
6-0	DPLL_DIV	R/W	0h	DPLL divider factor (0 to 127) (equal to input N of DPLL; actual division factor is N+1).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.4 CM_AUTOIDLE_DPLL_PER Register (offset = 6Ch)

This register provides automatic control over the DPLL activity.

CM_AUTOIDLE_DPLL_PER is shown in Figure 7-135 and described in Table 7-149.

Figure 7-135. CM_AUTOIDLE_DPLL_PER Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-149. CM_AUTOIDLE_DPLL_PER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	Reserved	R	0h	
2-0	AUTO_DPLL_MODE	R/W	0h	DPLL automatic control; 0 - AUTO_CTL_DISABLE: DPLL auto control disabled 1 - AUTO_LP_STOP: The DPLL is automatically put in Low Power Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 2 - Reserved2: Reserved 3 - Reserved3: Reserved 4 - Reserved4: Reserved 5 - AUTO_LP_BYP: The DPLL is automatically put in Idle Bypass Low Power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 6 - Reserved6: Reserved 7 - Reserved7: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.5 CM_IDLEST_DPLL_PER Register (offset = 70h)

This register allows monitoring the master clock activity. This register is read only and automatically updated. [warm reset insensitive]

CM_IDLEST_DPLL_PER is shown in Figure 7-136 and described in Table 7-150.

Figure 7-136. CM_IDLEST_DPLL_PER Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-150. CM_IDLEST_DPLL_PER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R	0h	
8	ST_MN_BYPASS	R	0h	DPLL MN_BYPASS status 0 - NO_MNBYPASS: DPLL is not in MN_Bypass 1 - MN_BYPASS: DPLL is in MN_Bypass
7-1	Reserved	R	0h	
0	ST_DPLL_CLK	R	0h	DPLL clock activity 0 - DPLL_UNLOCKED: DPLL is either in bypass mode or in stop mode. 1 - DPLL_LOCKED: DPLL is LOCKED

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.6 CM_SSC_DELTAMSTEP_DPLL_PER Register (offset = 74h)

Control the DeltaMStep parameter for Spread Spectrum Clocking technique DeltaMStep is split into fractional and integer part. [warm reset insensitive]

CM_SSC_DELTAMSTEP_DPLL_PER is shown in Figure 7-137 and described in Table 7-151.

Figure 7-137. CM_SSC_DELTAMSTEP_DPLL_PER Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-151. CM_SSC_DELTAMSTEP_DPLL_PER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	Reserved	R	0h	
19-0	DELTAMSTEP	R/W	0h	Fractional setting for DeltaMStep parameter

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.7 CM_WKUP_L4WKUP_CLKCTRL Register (offset = Ch)

This register manages the L4WKUP clocks.

CM_WKUP_L4WKUP_CLKCTRL is shown in Figure 7-138 and described in Table 7-152.

Figure 7-138. CM_WKUP_L4WKUP_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-152. CM_WKUP_L4WKUP_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	Reserved	R	0h	
17-16	IDLEST	R	0h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R	2h	Control the way mandatory clocks are managed.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.8 CM_SSC_MODFREQDIV_DPLL_PER Register (offset = 78h)

Control the Modulation Frequency (Fm) for Spread Spectrum Clocking technique by defining it as a ratio of DPLL_REFCLK/4 Fm = [DPLL_REFCLK/4]/MODFREQDIV MODFREQDIV = MODFREQDIV_MANTISSA * 2^MODFREQDIV_EXPONENT [warm reset insensitive]

CM_SSC_MODFREQDIV_DPLL_PER is shown in Figure 7-139 and described in Table 7-153.

Figure 7-139. CM_SSC_MODFREQDIV_DPLL_PER Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-153. CM_SSC_MODFREQDIV_DPLL_PER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	Reserved	R	0h	
10-8	MODFREQDIV_EXPONENT	R/W	0h	Set the Exponent component of MODFREQDIV factor
7	Reserved	R	0h	
6-0	MODFREQDIV_MANTISSA	R/W	0h	Set the Mantissa component of MODFREQDIV factor

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.9 CM_CLKDCOLDO_DPLL_PER Register (offset = 7Ch)

This register provides controls over the M2 divider of the DPLL.

CM_CLKDCOLDO_DPLL_PER is shown in Figure 7-140 and described in Table 7-154.

Figure 7-140. CM_CLKDCOLDO_DPLL_PER Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-154. CM_CLKDCOLDO_DPLL_PER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	Reserved	R	0h	
12	DPLL_CLKDCOLDO_PWDN	R/W	0h	Automatic power down for CLKDCOLDO o/p when it is gated 0 - ALWAYS_ACTIVE: Keep CLKDCOLDO o/p powered on even when it is gated 1 - AUTO_PWDN: Automatically power down CLKDCOLDO o/p when it is gated.
11-10	Reserved	R	0h	
9	ST_DPLL_CLKDCOLDO	R	0h	DPLL CLKDCOLDO status 0 - CLK_ENABLED: The clock output is enabled 1 - CLK_GATED: The clock output is gated
8	DPLL_CLKDCOLDO_GATE_CTRL	R/W	0h	Control gating of DPLL CLKDCOLDO 0 - CLK_AUTOGATE: Automatically gate this clock when there is no dependency for it 1 - CLK_ENABLE: Force this clock to stay enabled even if there is no request
7-0	Reserved	R	0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.10 CM_DIV_M4_DPLL_CORE Register (offset = 80h)

This register provides controls over the CLKOUT1 o/p of the HSDIVIDER.

CM_DIV_M4_DPLL_CORE is shown in Figure 7-141 and described in Table 7-155.

Figure 7-141. CM_DIV_M4_DPLL_CORE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-155. CM_DIV_M4_DPLL_CORE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	Reserved	R	0h	
12	HSDIVIDER_CLKOUT1_PWDN	R/W	0h	Automatic power down for HSDIVIDER M4 divider and hence CLKOUT1 output when the o/p clock is gated. 0 - ALWAYS_ACTIVE: Keep M4 divider powered on even when CLKOUT1 is gated. 1 - AUTO_PWDN: Automatically power down M4 divider when CLKOUT1 is gated.
11-10	Reserved	R	0h	
9	ST_HSDIVIDER_CLKOUT1	R	0h	HSDIVIDER CLKOUT1 status 0 - CLK_ENABLED: The clock output is enabled 1 - CLK_GATED: The clock output is gated
8	HSDIVIDER_CLKOUT1_GATE_CTRL	R/W	0h	Control gating of HSDIVIDER CLKOUT1 0 - CLK_AUTOGATE: Automatically gate this clock when there is no dependency for it 1 - CLK_ENABLE: Force this clock to stay enabled even if there is no request
7-6	Reserved	R	0h	
5	HSDIVIDER_CLKOUT1_DIVCHACK	R	0h	Toggle on this status bit after changing HSDIVIDER_CLKOUT1_DIV indicates that the change in divider value has taken effect
4-0	HSDIVIDER_CLKOUT1_DIV	R/W	4h	DPLL post-divider factor, M4, for internal clock generation. Divide values from 1 to 31. 0 - Reserved: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.11 CM_DIV_M5_DPLL_CORE Register (offset = 84h)

This register provides controls over the CLKOUT2 o/p of the HSDIVIDER.

CM_DIV_M5_DPLL_CORE is shown in Figure 7-142 and described in Table 7-156.

Figure 7-142. CM_DIV_M5_DPLL_CORE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-156. CM_DIV_M5_DPLL_CORE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	Reserved	R	0h	
12	HSDIVIDER_CLKOUT2_PWDN	R/W	0h	Automatic power down for HSDIVIDER M5 divider and hence CLKOUT2 output when the o/p clock is gated. 0 - ALWAYS_ACTIVE: Keep M5 divider powered on even when CLKOUT2 is gated. 1 - AUTO_PWDN: Automatically power down M5 divider when CLKOUT2 is gated.
11-10	Reserved	R	0h	
9	ST_HSDIVIDER_CLKOUT2	R	0h	HSDIVIDER CLKOUT2 status 0 - CLK_ENABLED: The clock output is enabled 1 - CLK_GATED: The clock output is gated
8	HSDIVIDER_CLKOUT2_GATE_CTRL	R/W	0h	Control gating of HSDIVIDER CLKOUT2 0 - CLK_AUTOGATE: Automatically gate this clock when there is no dependency for it 1 - CLK_ENABLE: Force this clock to stay enabled even if there is no request
7-6	Reserved	R	0h	
5	HSDIVIDER_CLKOUT2_DIVCHACK	R	0h	Toggle on this status bit after changing HSDIVIDER_CLKOUT2_DIV indicates that the change in divider value has taken effect
4-0	HSDIVIDER_CLKOUT2_DIV	R/W	4h	DPLL post-divider factor, M5, for internal clock generation. Divide values from 1 to 31. 0 - Reserved: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.12 CM_CLKMODE_DPLL_MPU Register (offset = 88h)

This register allows controlling the DPLL modes.

CM_CLKMODE_DPLL_MPU is shown in Figure 7-143 and described in Table 7-157.

Figure 7-143. CM_CLKMODE_DPLL_MPU Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-157. CM_CLKMODE_DPLL_MPU Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	Reserved	R	0h	
15	DPLL_SSC_TYPE	R/W	0h	Select between Triangular and SquareWave Spread Spectrum Clocking 0 - Triangular: Triangular Spread Spectrum Clocking is selected 1 - Square: Square Wave Spread Spectrum Clocking is selected (only available under proper licensing agreement)
14	DPLL_SSC_DOWNSPREAD	R/W	0h	Control if only low frequency spread is required 0 - FULL_SPREAD: When SSC is enabled, clock frequency is spread on both sides of the programmed frequency 1 - LOW_SPREAD: When SSC is enabled, clock frequency is spread only on the lower side of the programmed frequency
13	DPLL_SSC_ACK	R	0h	Acknowledgement from the DPLL regarding start and stop of Spread Spectrum Clocking feature 0 - Disabled: SSC has been turned off on PLL o/ps 1 - Enabled: SSC has been turned on on PLL o/ps
12	DPLL_SSC_EN	R/W	0h	Enable or disable Spread Spectrum Clocking 0 - Disabled: SSC disabled 1 - Enabled: SSC enabled
11	DPLL_REGM4XEN	R	0h	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. 0 - DISABLED: REGM4XEN mode of the DPLL is disabled
10	DPLL_LPMODE_EN	R/W	0h	Set the DPLL in Low Power mode. Check the DPLL documentation to see when this can be enabled. 0 - DISABLED: Low power mode of the DPLL is disabled 1 - ENABLED: Low power mode of the DPLL is enabled
9	DPLL_RELOCK_RAMP_EN	R/W	0h	If enabled, the clock ramping feature is used applied during the lock process, as well as the relock process. If disabled, the clock ramping feature is used only during the first lock.
8	DPLL_DRIFTGUARD_EN	R/W	0h	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0 - Diasbled: DRIFTGUARD feature is disabled 1 - Enabled: DRIFTGUARD feature is enabled
7-5	DPLL_RAMP_RATE	R/W	0h	Selects the time in terms of DPLL REFCLKs spent at each stage of the clock ramping process 0 - REFCLKx2: 2 REFCLKs 1 - REFCLKx4: 4 REFCLKs 2 - REFCLKx8: 8 REFCLKs 3 - REFCLKx16: 16 REFCLKs 4 - REFCLKx32: 32 REFCLKs 5 - REFCLKx64: 64 REFCLKs 6 - REFCLKx128: 128 REFCLKs 7 - REFCLKx512: 512 REFCLKs

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 7-157. CM_CLKMODE_DPLL_MPU Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-3	DPLL_RAMP_LEVEL	R/W	0h	The DPLL provides an output clock frequency ramping feature when switching from bypass clock to normal clock during lock and re-lock. The frequency ramping will happen in a maximum of 4 steps in frequency before the DPLL's frequency lock indicator is asserted. This register is used to enable/disable the DPLL ramping feature. If enabled, it is also used to select the algorithm used for clock ramping 0 - RAMP_DISABLE: CLKOUT => No ramping CLKOUTX2 => No ramping 1 - RAMP_ALGO1: CLKOUT => Bypass clk -> Fout/8 -> Fout/4 -> Fout/2 -> Fout CLKOUTX2 => Bypass clk -> Foutx2/8 -> Foutx2/4 -> Foutx2/2 -> Foutx2 2 - RAMP_ALGO2: CLKOUT => Bypass clk -> Fout/4 -> Fout/2 -> Fout/1.5 -> Fout CLKOUTX2 => Bypass clk -> Foutx2/4 -> Foutx2/2 -> Foutx2/1.5 -> Foutx2 3 - Reserved: Reserved
2-0	DPLL_EN	R/W	4h	DPLL control. Upon Warm Reset, the PRCM DPLL control state machine updates this register to reflect MN Bypass mode. 0 - Reserved: Reserved 1 - Reserved1: Reserved 2 - Reserved2: Reserved 3 - Reserved3: Reserved 4 - DPLL_MN_BYP_MODE: Put the DPLL in MN Bypass mode. The DPLL_MULT register bits are reset to 0 automatically by putting the DPLL in this mode. 5 - DPLL_LP_BYP_MODE: Put the DPLL in Idle Bypass Low Power mode. 6 - DPLL_FR_BYP_MODE: Put the DPLL in Idle Bypass Fast Relock mode. 7 - DPLL_LOCK_MODE: Enables the DPLL in Lock mode

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.13 CM_CLKMODE_DPLL_PER Register (offset = 8Ch)

This register allows controlling the DPLL modes.

CM_CLKMODE_DPLL_PER is shown in Figure 7-144 and described in Table 7-158.

Figure 7-144. CM_CLKMODE_DPLL_PER Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-158. CM_CLKMODE_DPLL_PER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	Reserved	R	0h	
15	DPLL_SSC_TYPE	R/W	0h	Select between Triangular and SquareWave Spread Spectrum Clocking 0 - Triangular: Triangular Spread Spectrum Clocking is selected 1 - Square: Square Wave Spread Spectrum Clocking is selected (only available under proper licensing agreement)
14	DPLL_SSC_DOWNSPREAD	R/W	0h	Control if only low frequency spread is required 0 - FULL_SPREAD: When SSC is enabled, clock frequency is spread on both sides of the programmed frequency 1 - LOW_SPREAD: When SSC is enabled, clock frequency is spread only on the lower side of the programmed frequency
13	DPLL_SSC_ACK	R	0h	Acknowledgement from the DPLL regarding start and stop of Spread Spectrum Clocking feature 0 - Disabled: SSC has been turned off on PLL o/ps 1 - Enabled: SSC has been turned on on PLL o/ps
12	DPLL_SSC_EN	R/W	0h	Enable or disable Spread Spectrum Clocking 0 - Disabled: SSC disabled 1 - Enabled: SSC enabled
11-3	Reserved	R	0h	
2-0	DPLL_EN	R/W	4h	DPLL control. Upon Warm Reset, the PRCM DPLL control state machine updates this register to reflect DPLL Low Power Stop mode. 0 - Reserved: Reserved 1 - DPLL_LP_STP_MODE: Put the DPLL in Low Power Stop mode 2 - Reserved2: Reserved2 3 - Reserved3: Reserved 4 - DPLL_MN_BYP_MODE: Put the DPLL in MN Bypass mode. The DPLL_MULT register bits are reset to 0 automatically by putting the DPLL in this mode. 5 - DPLL_LP_BYP_MODE: Put the DPLL in Idle Bypass Low Power mode. 6 - Reserved6: Reserved 7 - DPLL_LOCK_MODE: Enables the DPLL in Lock mode

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.14 CM_CLKMODE_DPLL_CORE Register (offset = 90h)

This register allows controlling the DPLL modes.

CM_CLKMODE_DPLL_CORE is shown in Figure 7-145 and described in Table 7-159.

Figure 7-145. CM_CLKMODE_DPLL_CORE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-159. CM_CLKMODE_DPLL_CORE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	Reserved	R	0h	
15	DPLL_SSC_TYPE	R/W	0h	Select between Triangular and SquareWave Spread Spectrum Clocking 0 - Triangular: Triangular Spread Spectrum Clocking is selected 1 - Square: Square Wave Spread Spectrum Clocking is selected (only available under proper licensing agreement)
14	DPLL_SSC_DOWNSPREAD	R/W	0h	Control if only low frequency spread is required 0 - FULL_SPREAD: When SSC is enabled, clock frequency is spread on both sides of the programmed frequency 1 - LOW_SPREAD: When SSC is enabled, clock frequency is spread only on the lower side of the programmed frequency
13	DPLL_SSC_ACK	R	0h	Acknowledgement from the DPLL regarding start and stop of Spread Spectrum Clocking feature 0 - Disabled: SSC has been turned off on PLL o/ps 1 - Enabled: SSC has been turned on on PLL o/ps
12	DPLL_SSC_EN	R/W	0h	Enable or disable Spread Spectrum Clocking 0 - Disabled: SSC disabled 1 - Enabled: SSC enabled
11	DPLL_REGM4XEN	R	0h	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. 0 - DISABLED: REGM4XEN mode of the DPLL is disabled
10	DPLL_LPMODE_EN	R/W	0h	Set the DPLL in Low Power mode. Check the DPLL documentation to see when this can be enabled. 0 - DISABLED: Low power mode of the DPLL is disabled 1 - ENABLED: Low power mode of the DPLL is enabled
9	DPLL_RELOCK_RAMP_EN	R/W	0h	If enabled, the clock ramping feature is used applied during the lock process, as well as the relock process. If disabled, the clock ramping feature is used only during the first lock.
8	DPLL_DRIFTGUARD_EN	R/W	0h	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0 - Diasbled: DRIFTGUARD feature is disabled 1 - Enabled: DRIFTGUARD feature is enabled
7-5	DPLL_RAMP_RATE	R/W	0h	Selects the time in terms of DPLL REFCLKs spent at each stage of the clock ramping process 0 - REFCLKx2: 2 REFCLKs 1 - REFCLKx4: 4 REFCLKs 2 - REFCLKx8: 8 REFCLKs 3 - REFCLKx16: 16 REFCLKs 4 - REFCLKx32: 32 REFCLKs 5 - REFCLKx64: 64 REFCLKs 6 - REFCLKx128: 128 REFCLKs 7 - REFCLKx512: 512 REFCLKs

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 7-159. CM_CLKMODE_DPLL_CORE Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-3	DPLL_RAMP_LEVEL	R/W	0h	The DPLL provides an output clock frequency ramping feature when switching from bypass clock to normal clock during lock and re-lock. The frequency ramping will happen in a maximum of 4 steps in frequency before the DPLL's frequency lock indicator is asserted. This register is used to enable/disable the DPLL ramping feature. If enabled, it is also used to select the algorithm used for clock ramping 0 - RAMP_DISABLE: CLKOUT => No ramping CLKOUTX2 => No ramping 1 - RAMP_ALGO1: CLKOUT => Bypass clk -> Fout/8 -> Fout/4 -> Fout/2 -> Fout CLKOUTX2 => Bypass clk -> Foutx2/8 -> Foutx2/4 -> Foutx2/2 -> Foutx2 2 - RAMP_ALGO2: CLKOUT => Bypass clk -> Fout/4 -> Fout/2 -> Fout/1.5 -> Fout CLKOUTX2 => Bypass clk -> Foutx2/4 -> Foutx2/2 -> Foutx2/1.5 -> Foutx2 3 - Reserved: Reserved
2-0	DPLL_EN	R/W	4h	DPLL control. Upon Warm Reset, the PRCM DPLL control state machine updates this register to reflect MN Bypass mode. 0 - Reserved: Reserved 1 - Reserved1: Reserved 2 - Reserved2: Reserved 3 - Reserved3: Reserved 4 - DPLL_MN_BYP_MODE: Put the DPLL in MN Bypass mode. The DPLL_MULT register bits are reset to 0 automatically by putting the DPLL in this mode. 5 - DPLL_LP_BYP_MODE: Put the DPLL in Idle Bypass Low Power mode. 6 - DPLL_FR_BYP_MODE: Put the DPLL in Idle Bypass Fast Relock mode. 7 - DPLL_LOCK_MODE: Enables the DPLL in Lock mode

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.15 CM_CLKMODE_DPLL_DDR Register (offset = 94h)

This register allows controlling the DPLL modes.

CM_CLKMODE_DPLL_DDR is shown in Figure 7-146 and described in Table 7-160.

Figure 7-146. CM_CLKMODE_DPLL_DDR Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-160. CM_CLKMODE_DPLL_DDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	Reserved	R	0h	
15	DPLL_SSC_TYPE	R/W	0h	Select between Triangular and SquareWave Spread Spectrum Clocking 0 - Triangular: Triangular Spread Spectrum Clocking is selected 1 - Square: Square Wave Spread Spectrum Clocking is selected (only available under proper licensing agreement)
14	DPLL_SSC_DOWNSPREAD	R/W	0h	Control if only low frequency spread is required 0 - FULL_SPREAD: When SSC is enabled, clock frequency is spread on both sides of the programmed frequency 1 - LOW_SPREAD: When SSC is enabled, clock frequency is spread only on the lower side of the programmed frequency
13	DPLL_SSC_ACK	R	0h	Acknowledgement from the DPLL regarding start and stop of Spread Spectrum Clocking feature 0 - Disabled: SSC has been turned off on PLL o/ps 1 - Enabled: SSC has been turned on on PLL o/ps
12	DPLL_SSC_EN	R/W	0h	Enable or disable Spread Spectrum Clocking 0 - Disabled: SSC disabled 1 - Enabled: SSC enabled
11	DPLL_REGM4XEN	R	0h	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. 0 - DISABLED: REGM4XEN mode of the DPLL is disabled
10	DPLL_LPMODE_EN	R/W	0h	Set the DPLL in Low Power mode. Check the DPLL documentation to see when this can be enabled. 0 - DISABLED: Low power mode of the DPLL is disabled 1 - ENABLED: Low power mode of the DPLL is enabled
9	DPLL_RELOCK_RAMP_EN	R/W	0h	If enabled, the clock ramping feature is used applied during the lock process, as well as the relock process. If disabled, the clock ramping feature is used only during the first lock.
8	DPLL_DRIFTGUARD_EN	R/W	0h	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0 - Diasbled: DRIFTGUARD feature is disabled 1 - Enabled: DRIFTGUARD feature is enabled
7-5	DPLL_RAMP_RATE	R/W	0h	Selects the time in terms of DPLL REFCLKs spent at each stage of the clock ramping process 0 - REFCLKx2: 2 REFCLKs 1 - REFCLKx4: 4 REFCLKs 2 - REFCLKx8: 8 REFCLKs 3 - REFCLKx16: 16 REFCLKs 4 - REFCLKx32: 32 REFCLKs 5 - REFCLKx64: 64 REFCLKs 6 - REFCLKx128: 128 REFCLKs 7 - REFCLKx512: 512 REFCLKs

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 7-160. CM_CLKMODE_DPLL_DDR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-3	DPLL_RAMP_LEVEL	R/W	0h	The DPLL provides an output clock frequency ramping feature when switching from bypass clock to normal clock during lock and re-lock. The frequency ramping will happen in a maximum of 4 steps in frequency before the DPLL's frequency lock indicator is asserted. This register is used to enable/disable the DPLL ramping feature. If enabled, it is also used to select the algorithm used for clock ramping 0 - RAMP_DISABLE: CLKOUT => No ramping CLKOUTX2 => No ramping 1 - RAMP_ALGO1: CLKOUT => Bypass clk -> Fout/8 -> Fout/4 -> Fout/2 -> Fout CLKOUTX2 => Bypass clk -> Foutx2/8 -> Foutx2/4 -> Foutx2/2 -> Foutx2 2 - RAMP_ALGO2: CLKOUT => Bypass clk -> Fout/4 -> Fout/2 -> Fout/1.5 -> Fout CLKOUTX2 => Bypass clk -> Foutx2/4 -> Foutx2/2 -> Foutx2/1.5 -> Foutx2 3 - Reserved: Reserved
2-0	DPLL_EN	R/W	4h	DPLL control. Upon Warm Reset, the PRCM DPLL control state machine updates this register to reflect MN Bypass mode. 0 - Reserved: Reserved 1 - Reserved1: Reserved 2 - Reserved2: Reserved 3 - Reserved3: Reserved 4 - DPLL_MN_BYP_MODE: Put the DPLL in MN Bypass mode. The DPLL_MULT register bits are reset to 0 automatically by putting the DPLL in this mode. 5 - DPLL_LP_BYP_MODE: Put the DPLL in Idle Bypass Low Power mode. 6 - DPLL_FR_BYP_MODE: Put the DPLL in Idle Bypass Fast Relock mode. 7 - DPLL_LOCK_MODE: Enables the DPLL in Lock mode

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.16 CM_CLKMODE_DPLL_DISP Register (offset = 98h)

This register allows controlling the DPLL modes.

CM_CLKMODE_DPLL_DISP is shown in Figure 7-147 and described in Table 7-161.

Figure 7-147. CM_CLKMODE_DPLL_DISP Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-161. CM_CLKMODE_DPLL_DISP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	Reserved	R	0h	
15	DPLL_SSC_TYPE	R/W	0h	Select between Triangular and SquareWave Spread Spectrum Clocking 0 - Triangular: Triangular Spread Spectrum Clocking is selected 1 - Square: Square Wave Spread Spectrum Clocking is selected (only available under proper licensing agreement)
14	DPLL_SSC_DOWNSPREAD	R/W	0h	Control if only low frequency spread is required 0 - FULL_SPREAD: When SSC is enabled, clock frequency is spread on both sides of the programmed frequency 1 - LOW_SPREAD: When SSC is enabled, clock frequency is spread only on the lower side of the programmed frequency
13	DPLL_SSC_ACK	R	0h	Acknowledgement from the DPLL regarding start and stop of Spread Spectrum Clocking feature 0 - Disabled: SSC has been turned off on PLL o/ps 1 - Enabled: SSC has been turned on on PLL o/ps
12	DPLL_SSC_EN	R/W	0h	Enable or disable Spread Spectrum Clocking 0 - Disabled: SSC disabled 1 - Enabled: SSC enabled
11	DPLL_REGM4XEN	R	0h	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. 0 - DISABLED: REGM4XEN mode of the DPLL is disabled
10	DPLL_LPMODE_EN	R/W	0h	Set the DPLL in Low Power mode. Check the DPLL documentation to see when this can be enabled. 0 - DISABLED: Low power mode of the DPLL is disabled 1 - ENABLED: Low power mode of the DPLL is enabled
9	DPLL_RELOCK_RAMP_EN	R/W	0h	If enabled, the clock ramping feature is used applied during the lock process, as well as the relock process. If disabled, the clock ramping feature is used only during the first lock.
8	DPLL_DRIFTGUARD_EN	R/W	0h	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0 - Diasbled: DRIFTGUARD feature is disabled 1 - Enabled: DRIFTGUARD feature is enabled
7-5	DPLL_RAMP_RATE	R/W	0h	Selects the time in terms of DPLL REFCLKs spent at each stage of the clock ramping process 0 - REFCLKx2: 2 REFCLKs 1 - REFCLKx4: 4 REFCLKs 2 - REFCLKx8: 8 REFCLKs 3 - REFCLKx16: 16 REFCLKs 4 - REFCLKx32: 32 REFCLKs 5 - REFCLKx64: 64 REFCLKs 6 - REFCLKx128: 128 REFCLKs 7 - REFCLKx512: 512 REFCLKs

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 7-161. CM_CLKMODE_DPLL_DISP Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
4-3	DPLL_RAMP_LEVEL	R/W	0h	The DPLL provides an output clock frequency ramping feature when switching from bypass clock to normal clock during lock and re-lock. The frequency ramping will happen in a maximum of 4 steps in frequency before the DPLL's frequency lock indicator is asserted. This register is used to enable/disable the DPLL ramping feature. If enabled, it is also used to select the algorithm used for clock ramping 0 - RAMP_DISABLE: CLKOUT => No ramping CLKOUTX2 => No ramping 1 - RAMP_ALGO1: CLKOUT => Bypass clk -> Fout/8 -> Fout/4 -> Fout/2 -> Fout CLKOUTX2 => Bypass clk -> Foutx2/8 -> Foutx2/4 -> Foutx2/2 -> Foutx2 2 - RAMP_ALGO2: CLKOUT => Bypass clk -> Fout/4 -> Fout/2 -> Fout/1.5 -> Fout CLKOUTX2 => Bypass clk -> Foutx2/4 -> Foutx2/2 -> Foutx2/1.5 -> Foutx2 3 - Reserved: Reserved
2-0	DPLL_EN	R/W	4h	DPLL control. Upon Warm Reset, the PRCM DPLL control state machine updates this register to reflect MN Bypass mode. 0 - Reserved: Reserved 1 - Reserved1: Reserved 2 - Reserved2: Reserved 3 - Reserved3: Reserved 4 - DPLL_MN_BYP_MODE: Put the DPLL in MN Bypass mode. The DPLL_MULT register bits are reset to 0 automatically by putting the DPLL in this mode. 5 - DPLL_LP_BYP_MODE: Put the DPLL in Idle Bypass Low Power mode. 6 - DPLL_FR_BYP_MODE: Put the DPLL in Idle Bypass Fast Relock mode. 7 - DPLL_LOCK_MODE: Enables the DPLL in Lock mode

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.17 CM_CLKSEL_DPLL_PERIPH Register (offset = 9Ch)

This register provides controls over the DPLL.

CM_CLKSEL_DPLL_PERIPH is shown in Figure 7-148 and described in Table 7-162.

Figure 7-148. CM_CLKSEL_DPLL_PERIPH Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-162. CM_CLKSEL_DPLL_PERIPH Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	DPLL_SD_DIV	R/W	0h	Sigma-Delta divider select (2-255). This factor must be set by s/w to ensure optimum jitter performance. DPLL_SD_DIV = CEILING ([DPLL_MULT/(DPLL_DIV+1)] * CLKINP / 250), where CLKINP is the input clock of the DPLL in MHz). Must be set with M and N factors, and must not be changed once DPLL is locked. 0 - Reserved: Reserved 1 - Reserved1: Reserved
23	Reserved	R	0h	
22-20	Reserved	R	0h	
19-8	DPLL_MULT	R/W	0h	DPLL multiplier factor (2 to 4095). This register is automatically cleared to 0 when the DPLL_EN field in the *CLKMODE_DPLL* register is set to select MN Bypass mode. (equal to input M of DPLL; M=2 to 4095 => DPLL multiplies by M). 0 - 0: Reserved 1 - 1: Reserved
7-0	DPLL_DIV	R/W	0h	DPLL divider factor (0 to 255) (equal to input N of DPLL; actual division factor is N+1).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.18 CM_WKUP_TIMER0_CLKCTRL Register (offset = 10h)

This register manages the TIMER0 clocks.

CM_WKUP_TIMER0_CLKCTRL is shown in Figure 7-149 and described in Table 7-163.

Figure 7-149. CM_WKUP_TIMER0_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-163. CM_WKUP_TIMER0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	2h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.19 CM_DIV_M2_DPLL_DDR Register (offset = A0h)

This register provides controls over the M2 divider of the DPLL.

CM_DIV_M2_DPLL_DDR is shown in Figure 7-150 and described in Table 7-164.

Figure 7-150. CM_DIV_M2_DPLL_DDR Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-164. CM_DIV_M2_DPLL_DDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	Reserved	R	0h	
9	ST_DPLL_CLKOUT	R	0h	DPLL CLKOUT status 0 - CLK_GATED: The clock output is gated 1 - CLK_ENABLED: The clock output is enabled
8	DPLL_CLKOUT_GATE_CTRL	R/W	0h	Control gating of DPLL CLKOUT 0 - CLK_AUTOGATE: Automatically gate this clock when there is no dependency for it 1 - CLK_ENABLE: Force this clock to stay enabled even if there is no request
7-6	Reserved	R	0h	
5	DPLL_CLKOUT_DIVCHACK	R	0h	Toggle on this status bit after changing DPLL_CLKOUT_DIV indicates that the change in divider value has taken effect
4-0	DPLL_CLKOUT_DIV	R/W	1h	DPLL M2 post-divider factor (1 to 31). 0 - Reserved: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.20 CM_DIV_M2_DPLL_DISP Register (offset = A4h)

This register provides controls over the M2 divider of the DPLL.

CM_DIV_M2_DPLL_DISP is shown in Figure 7-151 and described in Table 7-165.

Figure 7-151. CM_DIV_M2_DPLL_DISP Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-165. CM_DIV_M2_DPLL_DISP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	Reserved	R	0h	
9	ST_DPLL_CLKOUT	R	0h	DPLL CLKOUT status 0 - CLK_GATED: The clock output is gated 1 - CLK_ENABLED: The clock output is enabled
8	DPLL_CLKOUT_GATE_CTRL	R/W	0h	Control gating of DPLL CLKOUT 0 - CLK_AUTOGATE: Automatically gate this clock when there is no dependency for it 1 - CLK_ENABLE: Force this clock to stay enabled even if there is no request
7-6	Reserved	R	0h	
5	DPLL_CLKOUT_DIVCHACK	R	0h	Toggle on this status bit after changing DPLL_CLKOUT_DIV indicates that the change in divider value has taken effect
4-0	DPLL_CLKOUT_DIV	R/W	1h	DPLL M2 post-divider factor (1 to 31). 0 - Reserved: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.21 CM_DIV_M2_DPLL_MPU Register (offset = A8h)

This register provides controls over the M2 divider of the DPLL.

CM_DIV_M2_DPLL_MPU is shown in Figure 7-152 and described in Table 7-166.

Figure 7-152. CM_DIV_M2_DPLL_MPU Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-166. CM_DIV_M2_DPLL_MPU Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	Reserved	R	0h	
9	ST_DPLL_CLKOUT	R	0h	DPLL CLKOUT status 0 - CLK_GATED: The clock output is gated 1 - CLK_ENABLED: The clock output is enabled
8	DPLL_CLKOUT_GATE_CTRL	R/W	0h	Control gating of DPLL CLKOUT 0 - CLK_AUTOGATE: Automatically gate this clock when there is no dependency for it 1 - CLK_ENABLE: Force this clock to stay enabled even if there is no request
7-6	Reserved	R	0h	
5	DPLL_CLKOUT_DIVCHACK	R	0h	Toggle on this status bit after changing DPLL_CLKOUT_DIV indicates that the change in divider value has taken effect
4-0	DPLL_CLKOUT_DIV	R/W	1h	DPLL M2 post-divider factor (1 to 31). 0 - Reserved: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.22 CM_DIV_M2_DPLL_PER Register (offset = ACh)

This register provides controls over the M2 divider of the DPLL.

CM_DIV_M2_DPLL_PER is shown in Figure 7-153 and described in Table 7-167.

Figure 7-153. CM_DIV_M2_DPLL_PER Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-167. CM_DIV_M2_DPLL_PER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	Reserved	R	0h	
9	ST_DPLL_CLKOUT	R	0h	DPLL CLKOUT status 0 - CLK_GATED: The clock output is gated 1 - CLK_ENABLED: The clock output is enabled
8	DPLL_CLKOUT_GATE_CTRL	R/W	0h	Control gating of DPLL CLKOUT 0 - CLK_AUTOGATE: Automatically gate this clock when there is no dependency for it 1 - CLK_ENABLE: Force this clock to stay enabled even if there is no request
7	DPLL_CLKOUT_DIVCHACK	R	0h	Toggle on this status bit after changing DPLL_CLKOUT_DIV indicates that the change in divider value has taken effect
6-0	DPLL_CLKOUT_DIV	R/W	1h	DPLL M2 post-divider factor (1 to 31). 0 - Reserved: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.23 CM_WKUP_WKUP_M3_CLKCTRL Register (offset = B0h)

This register manages the WKUP M3 clocks.

CM_WKUP_WKUP_M3_CLKCTRL is shown in Figure 7-154 and described in Table 7-168.

Figure 7-154. CM_WKUP_WKUP_M3_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-168. CM_WKUP_WKUP_M3_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	STBYST	R	0h	Module standby status. 0 - Func: Module is functional (not in standby) 1 - Standby: Module is in standby
17-2	Reserved	R	0h	
1-0	MODULEMODE	R	2h	Control the way mandatory clocks are managed.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.24 CM_WKUP_UART0_CLKCTRL Register (offset = B4h)

This register manages the UART0 clocks.

CM_WKUP_UART0_CLKCTRL is shown in Figure 7-155 and described in Table 7-169.

Figure 7-155. CM_WKUP_UART0_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-169. CM_WKUP_UART0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.25 CM_WKUP_I2C0_CLKCTRL Register (offset = B8h)

This register manages the I2C0 clocks.

CM_WKUP_I2C0_CLKCTRL is shown in Figure 7-156 and described in Table 7-170.

Figure 7-156. CM_WKUP_I2C0_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-170. CM_WKUP_I2C0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.26 CM_WKUP_ADC_TSC_CLKCTRL Register (offset = BCh)

This register manages the ADC clocks.

CM_WKUP_ADC_TSC_CLKCTRL is shown in Figure 7-157 and described in Table 7-171.

Figure 7-157. CM_WKUP_ADC_TSC_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-171. CM_WKUP_ADC_TSC_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.27 CM_WKUP_SMARTREFLEX0_CLKCTRL Register (offset = C0h)

This register manages the SmartReflex0 clocks.

CM_WKUP_SMARTREFLEX0_CLKCTRL is shown in Figure 7-158 and described in Table 7-172.

Figure 7-158. CM_WKUP_SMARTREFLEX0_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-172. CM_WKUP_SMARTREFLEX0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.28 CM_WKUP_TIMER1_CLKCTRL Register (offset = C4h)

This register manages the TIMER1 clocks.

CM_WKUP_TIMER1_CLKCTRL is shown in Figure 7-159 and described in Table 7-173.

Figure 7-159. CM_WKUP_TIMER1_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-173. CM_WKUP_TIMER1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.29 CM_WKUP_DEBUGSS_CLKCTRL Register (offset = 14h)

This register manages the DEBUGSS clocks.

CM_WKUP_DEBUGSS_CLKCTRL is shown in Figure 7-160 and described in Table 7-174.

Figure 7-160. CM_WKUP_DEBUGSS_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-174. CM_WKUP_DEBUGSS_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	Reserved	R	0h	
30	OPTCLK_DEBUG_CLKA	R/W	1h	
29-27	stm_pmd_clkdivsel	R/W	2h	
26-24	trc_pmd_clkdivsel	R/W	2h	
23-22	stm_pmd_clksel	R/W	1h	
21-20	trc_pmd_clksel	R/W	1h	
19	OPTFCLKEN_DBGSYSCLK	R/W	1h	Optional functional clock control. 0 - FCLK_DIS: Optional functional clock is disabled 1 - FCLK_EN: Optional functional clock is enabled
18	STBYST	R	0h	Module standby status. 0 - Func: Module is functional (not in standby) 1 - Standby: Module is in standby
17-16	IDLEST	R	0h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	2h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.30 CM_WKUP_SMARTREFLEX1_CLKCTRL Register (offset = C8h)

This register manages the SmartReflex1 clocks.

CM_WKUP_SMARTREFLEX1_CLKCTRL is shown in Figure 7-161 and described in Table 7-175.

Figure 7-161. CM_WKUP_SMARTREFLEX1_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-175. CM_WKUP_SMARTREFLEX1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.31 CM_L4_WKUP_AON_CLKSTCTRL Register (offset = CCh)

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

CM_L4_WKUP_AON_CLKSTCTRL is shown in Figure 7-162 and described in Table 7-176.

Figure 7-162. CM_L4_WKUP_AON_CLKSTCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-176. CM_L4_WKUP_AON_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	R	0h	
25-14	Reserved	R	0h	
13-8	Reserved	R	0h	
7-4	Reserved	R	0h	
3	Reserved	R	0h	
2	CLKACTIVITY_L4_WKUP_AON_GCLK	R	1h	This field indicates the state of the L4_WKUP clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
1-0	CLKTRCTRL	R	2h	Controls the clock state transition of the always on L4 clock domain.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.32 CM_WKUP_WDT0_CLKCTRL Register (offset = D0h)

This register manages the WDT0 clocks.

CM_WKUP_WDT0_CLKCTRL is shown in Figure 7-163 and described in Table 7-177.

Figure 7-163. CM_WKUP_WDT0_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-177. CM_WKUP_WDT0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	2h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.33 CM_WKUP_WDT1_CLKCTRL Register (offset = D4h)

This register manages the WDT1 clocks.

CM_WKUP_WDT1_CLKCTRL is shown in Figure 7-164 and described in Table 7-178.

Figure 7-164. CM_WKUP_WDT1_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-178. CM_WKUP_WDT1_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	2h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.34 CM_DIV_M6_DPLL_CORE Register (offset = D8h)

This register provides controls over the CLKOUT3 o/p of the HSDIVIDER. [warm reset insensitive]

CM_DIV_M6_DPLL_CORE is shown in Figure 7-165 and described in Table 7-179.

Figure 7-165. CM_DIV_M6_DPLL_CORE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-179. CM_DIV_M6_DPLL_CORE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-13	Reserved	R	0h	
12	HSDIVIDER_CLKOUT3_PWDN	R/W	0h	Automatic power down for HSDIVIDER M6 divider and hence CLKOUT3 output when the o/p clock is gated. 0 - ALWAYS_ACTIVE: Keep M6 divider powered on even when CLKOUT2 is gated. 1 - AUTO_PWDN: Automatically power down M6 divider when CLKOUT2 is gated.
11-10	Reserved	R	0h	
9	ST_HSDIVIDER_CLKOUT3	R	0h	HSDIVIDER CLKOUT3 status 0 - CLK_ENABLED: The clock output is enabled 1 - CLK_GATED: The clock output is gated
8	HSDIVIDER_CLKOUT3_GATE_CTRL	R/W	0h	Control gating of HSDIVIDER CLKOUT3 0 - CLK_AUTOGATE: Automatically gate this clock when there is no dependency for it 1 - CLK_ENABLE: Force this clock to stay enabled even if there is no request
7-6	Reserved	R	0h	
5	HSDIVIDER_CLKOUT3_DIVCHACK	R	0h	Toggle on this status bit after changing HSDIVIDER_CLKOUT3_DIV indicates that the change in divider value has taken effect
4-0	HSDIVIDER_CLKOUT3_DIV	R/W	4h	DPLL post-divider factor, M6, for internal clock generation. Divide values from 1 to 31. 0 - Reserved: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.35 CM_L3_AON_CLKSTCTRL Register (offset = 18h)

This register enables the domain power state transition. It controls the SW supervised clock domain state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.

CM_L3_AON_CLKSTCTRL is shown in Figure 7-166 and described in Table 7-180.

Figure 7-166. CM_L3_AON_CLKSTCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-180. CM_L3_AON_CLKSTCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved	R	0h	
25-11	Reserved	R	0h	
10-8	Reserved	R	0h	
7-5	Reserved	R	0h	
4	CLKACTIVITY_DEBUG_CLKA	R	1h	This field indicates the state of the Debugss CLKA clock in the domain.
3	CLKACTIVITY_L3_AON_GCLK	R	1h	This field indicates the state of the L3_AON clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
2	CLKACTIVITY_DBGSYSCLK	R	0h	This field indicates the state of the Debugss sysclk clock in the domain. 0 - Inact: Corresponding clock is gated 1 - Act: Corresponding clock is active
1-0	CLKTRCTRL	R/W	2h	Controls the clock state transition of the l3 AON clock domain. 0 - NO_SLEEP: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 1 - SW_SLEEP: SW_SLEEP: Start a software forced sleep transition on the domain. 2 - SW_WKUP: SW_WKUP: Start a software forced wake-up transition on the domain. 3 - HW_AUTO: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.36 CM_AUTOIDLE_DPLL_MPU Register (offset = 1Ch)

This register provides automatic control over the DPLL activity.

CM_AUTOIDLE_DPLL_MPU is shown in Figure 7-167 and described in Table 7-181.

Figure 7-167. CM_AUTOIDLE_DPLL_MPU Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-181. CM_AUTOIDLE_DPLL_MPU Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	Reserved	R	0h	
2-0	AUTO_DPLL_MODE	R/W	0h	DPLL automatic control; 0 - AUTO_CTL_DISABLE: DPLL auto control disabled 1 - AUTO_LP_STOP: The DPLL is automatically put in Low Power Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 2 - Reserved2: Reserved 3 - Reserved3: Reserved 4 - Reserved4: Reserved 5 - AUTO_LP_BYP: The DPLL is automatically put in Idle Bypass Low Power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 6 - Reserved6: Reserved 7 - Reserved7: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.37 CM_IDLEST_DPLL_MPU Register (offset = 20h)

This register allows monitoring the master clock activity. This register is read only and automatically updated.[warm reset insensitive]

CM_IDLEST_DPLL_MPU is shown in Figure 7-168 and described in Table 7-182.

Figure 7-168. CM_IDLEST_DPLL_MPU Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-182. CM_IDLEST_DPLL_MPU Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R	0h	
8	ST_MN_BYPASS	R	0h	DPLL MN_BYPASS status 0 - NO_MNBYPASS: DPLL is not in MN_Bypass 1 - MN_BYPASS: DPLL is in MN_Bypass
7-1	Reserved	R	0h	
0	ST_DPLL_CLK	R	0h	DPLL clock activity 0 - DPLL_UNLOCKED: DPLL is either in bypass mode or in stop mode. 1 - DPLL_LOCKED: DPLL is LOCKED

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.38 CM_SSC_DELTAMSTEP_DPLL_MPU Register (offset = 24h)

Control the DeltaMStep parameter for Spread Spectrum Clocking technique DeltaMStep is split into fractional and integer part. [warm reset insensitive]

CM_SSC_DELTAMSTEP_DPLL_MPU is shown in Figure 7-169 and described in Table 7-183.

Figure 7-169. CM_SSC_DELTAMSTEP_DPLL_MPU Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-183. CM_SSC_DELTAMSTEP_DPLL_MPU Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	Reserved	R	0h	
19-0	DELTAMSTEP	R/W	0h	Fractional setting for DeltaMStep parameter

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.39 CM_WKUP_CONTROL_CLKCTRL Register (offset = 4h)

This register manages the Control Module clocks.

CM_WKUP_CONTROL_CLKCTRL is shown in Figure 7-170 and described in Table 7-184.

Figure 7-170. CM_WKUP_CONTROL_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-184. CM_WKUP_CONTROL_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	Reserved	R	0h	
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.40 CM_SSC_MODFREQDIV_DPLL_MPU Register (offset = 28h)

Control the Modulation Frequency (Fm) for Spread Spectrum Clocking technique by defining it as a ratio of DPLL_REFCLK/4 Fm = [DPLL_REFCLK/4]/MODFREQDIV MODFREQDIV = MODFREQDIV_MANTISSA * 2^MODFREQDIV_EXPONENT [warm reset insensitive]

CM_SSC_MODFREQDIV_DPLL_MPU is shown in Figure 7-171 and described in Table 7-185.

Figure 7-171. CM_SSC_MODFREQDIV_DPLL_MPU Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-185. CM_SSC_MODFREQDIV_DPLL_MPU Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	Reserved	R	0h	
10-8	MODFREQDIV_EXPONENT	R/W	0h	Set the Exponent component of MODFREQDIV factor
7	Reserved	R	0h	
6-0	MODFREQDIV_MANTISSA	R/W	0h	Set the Mantissa component of MODFREQDIV factor

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.41 CM_CLKSEL_DPLL_MPU Register (offset = 2Ch)

This register provides controls over the DPLL.

CM_CLKSEL_DPLL_MPU is shown in Figure 7-172 and described in Table 7-186.

Figure 7-172. CM_CLKSEL_DPLL_MPU Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-186. CM_CLKSEL_DPLL_MPU Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	Reserved	R	0h	
23	DPLL_BYP_CLKSEL	R/W	0h	Selects CLKINP or CLKINPULOW as Bypass Clock 0 - Sel0: Selects CLKINP Clock as BYPASS Clock 1 - Sel1: Selects CLKINPULOW as Bypass Clock
22-19	Reserved	R	0h	
18-8	DPLL_MULT	R/W	0h	DPLL multiplier factor (2 to 2047). This register is automatically cleared to 0 when the DPLL_EN field in the *CLKMODE_DPLL* register is set to select MN Bypass mode. (equal to input M of DPLL; M=2 to 2047 => DPLL multiplies by M). 0 - 0: Reserved 1 - 1: Reserved
7	Reserved	R	0h	
6-0	DPLL_DIV	R/W	0h	DPLL divider factor (0 to 127) (equal to input N of DPLL; actual division factor is N+1).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.42 CM_AUTOIDLE_DPLL_DDR Register (offset = 30h)

This register provides automatic control over the DPLL activity.

CM_AUTOIDLE_DPLL_DDR is shown in Figure 7-173 and described in Table 7-187.

Figure 7-173. CM_AUTOIDLE_DPLL_DDR Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-187. CM_AUTOIDLE_DPLL_DDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	Reserved	R	0h	
2-0	AUTO_DPLL_MODE	R/W	0h	DPLL automatic control; 0 - AUTO_CTL_DISABLE: DPLL auto control disabled 1 - AUTO_LP_STOP: The DPLL is automatically put in Low Power Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 2 - Reserved2: Reserved 3 - Reserved3: Reserved 4 - Reserved4: Reserved 5 - AUTO_LP_BYP: The DPLL is automatically put in Idle Bypass Low Power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 6 - Reserved6: Reserved 7 - Reserved7: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.43 CM_IDLEST_DPLL_DDR Register (offset = 34h)

This register allows monitoring the master clock activity. This register is read only and automatically updated. [warm reset insensitive]

CM_IDLEST_DPLL_DDR is shown in Figure 7-174 and described in Table 7-188.

Figure 7-174. CM_IDLEST_DPLL_DDR Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-188. CM_IDLEST_DPLL_DDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R	0h	
8	ST_MN_BYPASS	R	0h	DPLL MN_BYPASS status 0 - NO_MNBYPASS: DPLL is not in MN_Bypass 1 - MN_BYPASS: DPLL is in MN_Bypass
7-1	Reserved	R	0h	
0	ST_DPLL_CLK	R	0h	DPLL clock activity 0 - DPLL_UNLOCKED: DPLL is either in bypass mode or in stop mode. 1 - DPLL_LOCKED: DPLL is LOCKED

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.44 CM_SSC_DELTAMSTEP_DPLL_DDR Register (offset = 38h)

Control the DeltaMStep parameter for Spread Spectrum Clocking technique DeltaMStep is split into fractional and integer part. [warm reset insensitive]

CM_SSC_DELTAMSTEP_DPLL_DDR is shown in Figure 7-175 and described in Table 7-189.

Figure 7-175. CM_SSC_DELTAMSTEP_DPLL_DDR Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-189. CM_SSC_DELTAMSTEP_DPLL_DDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	Reserved	R	0h	
19-0	DELTAMSTEP	R/W	0h	Fractional setting for DeltaMStep parameter

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.45 CM_SSC_MODFREQDIV_DPLL_DDR Register (offset = 3Ch)

Control the Modulation Frequency (Fm) for Spread Spectrum Clocking technique by defining it as a ratio of DPLL_REFCLK/4 Fm = [DPLL_REFCLK/4]/MODFREQDIV MODFREQDIV = MODFREQDIV_MANTISSA * 2^MODFREQDIV_EXPONENT [warm reset insensitive]

CM_SSC_MODFREQDIV_DPLL_DDR is shown in Figure 7-176 and described in Table 7-190.

Figure 7-176. CM_SSC_MODFREQDIV_DPLL_DDR Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-190. CM_SSC_MODFREQDIV_DPLL_DDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	Reserved	R	0h	
10-8	MODFREQDIV_EXPONENT	R/W	0h	Set the Exponent component of MODFREQDIV factor
7	Reserved	R	0h	
6-0	MODFREQDIV_MANTISSA	R/W	0h	Set the Mantissa component of MODFREQDIV factor

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.46 CM_CLKSEL_DPLL_DDR Register (offset = 40h)

This register provides controls over the DPLL.

CM_CLKSEL_DPLL_DDR is shown in Figure 7-177 and described in Table 7-191.

Figure 7-177. CM_CLKSEL_DPLL_DDR Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-191. CM_CLKSEL_DPLL_DDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	Reserved	R	0h	
23	DPLL_BYP_CLKSEL	R/W	0h	Select CLKINP orr CLKINPULOW as bypass clock 0 - Sel0: Selects CLKINP Clock as BYPASS Clock 1 - Sel1: Selects CLKINPULOW as Bypass Clock
22-19	Reserved	R	0h	
18-8	DPLL_MULT	R/W	0h	DPLL multiplier factor (2 to 2047). This register is automatically cleared to 0 when the DPLL_EN field in the *CLKMODE_DPLL* register is set to select MN Bypass mode. (equal to input M of DPLL; M=2 to 2047 => DPLL multiplies by M). 0 - 0: Reserved 1 - 1: Reserved
7	Reserved	R	0h	
6-0	DPLL_DIV	R/W	0h	DPLL divider factor (0 to 127) (equal to input N of DPLL; actual division factor is N+1).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.47 CM_AUTOIDLE_DPLL_DISP Register (offset = 44h)

This register provides automatic control over the DPLL activity.

CM_AUTOIDLE_DPLL_DISP is shown in Figure 7-178 and described in Table 7-192.

Figure 7-178. CM_AUTOIDLE_DPLL_DISP Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-192. CM_AUTOIDLE_DPLL_DISP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	Reserved	R	0h	
2-0	AUTO_DPLL_MODE	R/W	0h	DPLL automatic control; 0 - AUTO_CTL_DISABLE: DPLL auto control disabled 1 - AUTO_LP_STOP: The DPLL is automatically put in Low Power Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 2 - Reserved2: Reserved 3 - Reserved3: Reserved 4 - Reserved4: Reserved 5 - AUTO_LP_BYP: The DPLL is automatically put in Idle Bypass Low Power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 6 - Reserved6: Reserved 7 - Reserved7: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.48 CM_IDLEST_DPLL_DISP Register (offset = 48h)

This register allows monitoring the master clock activity. This register is read only and automatically updated. [warm reset insensitive]

CM_IDLEST_DPLL_DISP is shown in Figure 7-179 and described in Table 7-193.

Figure 7-179. CM_IDLEST_DPLL_DISP Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-193. CM_IDLEST_DPLL_DISP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R	0h	
8	ST_MN_BYPASS	R	0h	DPLL MN_BYPASS status 0 - NO_MNBYPASS: DPLL is not in MN_Bypass 1 - MN_BYPASS: DPLL is in MN_Bypass
7-1	Reserved	R	0h	
0	ST_DPLL_CLK	R	0h	DPLL clock activity 0 - DPLL_UNLOCKED: DPLL is either in bypass mode or in stop mode. 1 - DPLL_LOCKED: DPLL is LOCKED

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.49 CM_SSC_DELTAMSTEP_DPLL_DISP Register (offset = 4Ch)

Control the DeltaMStep parameter for Spread Spectrum Clocking technique DeltaMStep is split into fractional and integer part. [warm reset insensitive]

CM_SSC_DELTAMSTEP_DPLL_DISP is shown in Figure 7-180 and described in Table 7-194.

Figure 7-180. CM_SSC_DELTAMSTEP_DPLL_DISP Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-194. CM_SSC_DELTAMSTEP_DPLL_DISP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	Reserved	R	0h	
19-0	DELTAMSTEP	R/W	0h	Fractional setting for DeltaMStep parameter

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.50 CM_WKUP_GPIO0_CLKCTRL Register (offset = 8h)

This register manages the GPIO0 clocks.

CM_WKUP_GPIO0_CLKCTRL is shown in Figure 7-181 and described in Table 7-195.

Figure 7-181. CM_WKUP_GPIO0_CLKCTRL Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-195. CM_WKUP_GPIO0_CLKCTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved	R	0h	
18	OPTFCLKEN_GPIO0_GDBCLK	R/W	0h	Optional functional clock control. 0 - FCLK_DIS: Optional functional clock is disabled 1 - FCLK_EN: Optional functional clock is enabled
17-16	IDLEST	R	3h	Module idle status. 0 - Func: Module is fully functional, including OCP 1 - Trans: Module is performing transition: wakeup, or sleep, or sleep abortion 2 - Idle: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 3 - Disable: Module is disabled and cannot be accessed
15-2	Reserved	R	0h	
1-0	MODULEMODE	R/W	0h	Control the way mandatory clocks are managed. 0 - DISABLED: Module is disable by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 1 - RESERVED_1: Reserved 2 - ENABLE: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guarantied to stay present. As long as in this configuration, power domain sleep transition cannot happen. 3 - RESERVED: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.51 CM_SSC_MODFREQDIV_DPLL_DISP Register (offset = 50h)

Control the Modulation Frequency (Fm) for Spread Spectrum Clocking technique by defining it as a ratio of DPLL_REFCLK/4 Fm = [DPLL_REFCLK/4]/MODFREQDIV MODFREQDIV = MODFREQDIV_MANTISSA * 2^MODFREQDIV_EXPONENT [warm reset insensitive]

CM_SSC_MODFREQDIV_DPLL_DISP is shown in Figure 7-182 and described in Table 7-196.

Figure 7-182. CM_SSC_MODFREQDIV_DPLL_DISP Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-196. CM_SSC_MODFREQDIV_DPLL_DISP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	Reserved	R	0h	
10-8	MODFREQDIV_EXPONENT	R/W	0h	Set the Exponent component of MODFREQDIV factor
7	Reserved	R	0h	
6-0	MODFREQDIV_MANTISSA	R/W	0h	Set the Mantissa component of MODFREQDIV factor

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.52 CM_CLKSEL_DPLL_DISP Register (offset = 54h)

This register provides controls over the DPLL.

CM_CLKSEL_DPLL_DISP is shown in Figure 7-183 and described in Table 7-197.

Figure 7-183. CM_CLKSEL_DPLL_DISP Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-197. CM_CLKSEL_DPLL_DISP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	Reserved	R	0h	
23	DPLL_BYP_CLKSEL	R/W	0h	Select CLKINP or CLKINPULOW as bypass clock 0 - Sel0: Selects CLKINP Clock as BYPASS Clock 1 - Sel1: Selects CLKINPULOW as Bypass Clock
22-19	Reserved	R	0h	
18-8	DPLL_MULT	R/W	0h	DPLL multiplier factor (2 to 2047). This register is automatically cleared to 0 when the DPLL_EN field in the *CLKMODE_DPLL* register is set to select MN Bypass mode. (equal to input M of DPLL; M=2 to 2047 => DPLL multiplies by M). 0 - 0: Reserved 1 - 1: Reserved
7	Reserved	R	0h	
6-0	DPLL_DIV	R/W	0h	DPLL divider factor (0 to 127) (equal to input N of DPLL; actual division factor is N+1).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.53 CM_AUTOIDLE_DPLL_CORE Register (offset = 58h)

This register provides automatic control over the DPLL activity.

CM_AUTOIDLE_DPLL_CORE is shown in Figure 7-184 and described in Table 7-198.

Figure 7-184. CM_AUTOIDLE_DPLL_CORE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 7-198. CM_AUTOIDLE_DPLL_CORE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	Reserved	R	0h	
2-0	AUTO_DPLL_MODE	R/W	0h	DPLL automatic control; 0 - AUTO_CTL_DISABLE: DPLL auto control disabled 1 - AUTO_LP_STOP: The DPLL is automatically put in Low Power Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 2 - Reserved2: Reserved 3 - Reserved3: Reserved 4 - Reserved4: Reserved 5 - AUTO_LP_BYP: The DPLL is automatically put in Idle Bypass Low Power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 6 - Reserved6: Reserved 7 - Reserved7: Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.54 CM_IDLEST_DPLL_CORE Register (offset = 5Ch)

This register allows monitoring the master clock activity. This register is read only and automatically updated. [warm reset insensitive]

CM_IDLEST_DPLL_CORE is shown in Figure 7-185 and described in Table 7-199.

Figure 7-185. CM_IDLEST_DPLL_CORE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-199. CM_IDLEST_DPLL_CORE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R	0h	
8	ST_MN_BYPASS	R	0h	DPLL MN_BYPASS status 0 - NO_MNBYPASS: DPLL is not in MN_Bypass 1 - MN_BYPASS: DPLL is in MN_Bypass
7-1	Reserved	R	0h	
0	ST_DPLL_CLK	R	0h	DPLL clock activity 0 - DPLL_UNLOCKED: DPLL is either in bypass mode or in stop mode. 1 - DPLL_LOCKED: DPLL is LOCKED

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

7.2.5.0.12.55 CM_SSC_DELTAMSTEP_DPLL_CORE Register (offset = 60h)

Control the DeltaMStep parameter for Spread Spectrum Clocking technique DeltaMStep is split into fractional and integer part. [warm reset insensitive]

CM_SSC_DELTAMSTEP_DPLL_CORE is shown in Figure 7-186 and described in Table 7-200.

Figure 7-186. CM_SSC_DELTAMSTEP_DPLL_CORE Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 7-200. CM_SSC_DELTAMSTEP_DPLL_CORE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	Reserved	R	0h	
19-0	DELTAMSTEP	R/W	0h	Fractional setting for DeltaMStep parameter

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Chapter 8

SPRUH73–July 2011

Control Module

This chapter describes the control module of the device.

Topic	Page

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Chapter 9

SPRUH73–July 2011

Interconnects

This chapter describes the interconnects of the device.

Topic **Page**

9.1	Introduction	1296
9.2	Integration	1296
9.3	Functional Description	1296
9.4	Use Cases	1296
9.5	Interconnect Registers	1297

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.1 Introduction

9.2 Integration

9.3 Functional Description

9.4 Use Cases

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5 Interconnect Registers

9.5.1 L3 Interconnect Registers

Table 9-1 lists the base address and address space for all L3 register blocks.

Table 9-1. Instance Summary

Module Name	Base Address	Size
RT	0x6800 0000	1K byte
SI	0x6800 0400	1K byte
IA_MPUSS	0x6800 1400	1K byte
IA_IVA2.2	0x6800 1800	1K byte
IA_SGX	0x6800 1C00	1K byte
TA_SMS	0x6800 2000	1K byte
TA_GPMC	0x6800 2400	1K byte
TA_OCM_RAM	0x6800 2800	1K byte
TA_OCM_ROM	0x6800 2C00	1K byte
IA_SAD2D	0x6800 3000	1K byte
TA_MAD2D	0x6800 3400	1K byte
IA_USB_HS_Host	0x6800 4000	1K byte
IA_USB_HS_OTG	0x6800 4400	1K byte
IA_sDMA_RD	0x6800 4C00	1K byte
IA_sDMA_WR	0x6800 5000	1K byte
IA_DSS	0x6800 5400	1K byte
IA_CAM	0x6800 5800	1K byte
IA_DAP	0x6800 5C00	1K byte
TA_IVA2.2	0x6800 6000	1K byte
TA_SGX	0x6800 6400	1K byte
TA_L4_CORE	0x6800 6800	1K byte
TA_L4_PER	0x6800 6C00	1K byte
TA_L4_EMU	0x6800 7000	1K byte
PM_RT	0x6801 0000	1K byte
PM_GPMC	0x6801 2400	1K byte
PM_OCM_RAM	0x6801 2800	1K byte
PM_OCM_ROM	0x6801 2C00	1K byte
PM_IVA2.2	0x6801 4000	1K byte

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.1.1 L3 Initiator Agent (L3 IA) Register Mapping Summary

This section describes the IA register block. Each IA in L3 interconnect has its own IA register block.

The following are the IA registers:

- MPU subsystem port
- IVA2.2 subsystem port
- SGX subsystem port
- High-Speed USB Host
- High-Speed USB OTG
- System DMA read port
- System DMA write port
- Display subsystem port
- Camera subsystem port
- DAP port
- Die-to-die

Table 9-2 through Table 9-5 list the IA registers and their physical addresses, depending on the module instance.

Table 9-2. Initiator Agent Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	IA_MPUSS Physical Address	IA_IVA2.2 Physical Address	SGX Physical Address	Section
L3_IA_AGENT_CONTROL	RW	64	0x020	0x6800 1420	0x6800 1820	0x6800 1C20	Section 9.5.1.2.1
L3_IA_AGENT_STATUS	RW	64	0x028	0x6800 1428	0x6800 1828	0x6800 1C28	Section 9.5.1.2.2
L3_IA_ERROR_LOG	RW	64	0x058	0x6800 1458	0x6800 1858	0x6800 1C58	Section 9.5.1.2.3
L3_IA_ERROR_LOG_ADDR	R	64	0x060	0x6800 1460	0x6800 1860	0x6800 1C60	Section 9.5.1.2.4

Table 9-3. Initiator Agent Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	IA_USB_HS_Host Physical Address	IA_USB_HS_OTG Physical Address	IA_SAD2D Physical Address	Section
L3_IA_AGENT_CONTROL	RW	64	0x020	0x6800 4020	0x6800 4420	0x6800 3020	Section 9.5.1.2.1
L3_IA_AGENT_STATUS	RW	64	0x028	0x6800 4028	0x6800 4428	0x6800 3028	Section 9.5.1.2.2
L3_IA_ERROR_LOG	RW	64	0x058	0x6800 4058	0x6800 4458	0x6800 3058	Section 9.5.1.2.3
L3_IA_ERROR_LOG_ADDR	R	64	0x060	0x6800 4060	0x6800 4460	0x6800 3060	Section 9.5.1.2.4

Table 9-4. Initiator Agent Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	IA_DMA_RD Physical Address	IA_DMA_WR Physical Address	Section
L3_IA_AGENT_CONTROL	RW	64	0x020	0x6800 4C20	0x6800 5020	Section 9.5.1.2.1
L3_IA_AGENT_STATUS	RW	64	0x028	0x6800 4C28	0x6800 5028	Section 9.5.1.2.2
L3_IA_ERROR_LOG	RW	64	0x058	0x6800 4C58	0x6800 5058	Section 9.5.1.2.3
L3_IA_ERROR_LOG_ADDR	R	64	0x060	0x6800 4C60	0x6800 5060	Section 9.5.1.2.4

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 9-5. Initiator Agent Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	IA_DSS Physical Address	IA_CAM Physical Address	IA_DAP Physical Address	Section
L3_IA_AGENT_CONTROL	RW	64	0x020	0x6800 5420	0x6800 5820	0x6800 5C20	Section 9.5.1.2.1
L3_IA_AGENT_STATUS	RW	64	0x028	0x6800 5428	0x6800 5828	0x6800 5C28	Section 9.5.1.2.2
L3_IA_ERROR_LOG	RW	64	0x058	0x6800 5458	0x6800 5858	0x6800 5C58	Section 9.5.1.2.3
L3_IA_ERROR_LOG_ADDR	R	64	0x060	0x6800 5460	0x6800 5860	0x6800 5C60	Section 9.5.1.2.4

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.1.2 L3 Initiator Agent (L3 IA) Register Descriptions

9.5.1.2.1 Agent Control Register - IA Block (L3_IA_AGENT_CONTROL)

The agent control register - IA block (L3_IA_AGENT_CONTROL) is shown in Figure 9-1 and described inTable 9-6.

Figure 9-1. Agent Control Register - IA Block (L3_IA_AGENT_CONTROL)

63–48	47–32	31–30	29	28	27	26	25	24
Reserved	Reserved	Reserved	NBAND_ERROR_SECONDARY_REP	INBAND_ERROR_PRIMARY_REP	ALL_INBAND_ERROR_REP	BURST_TIMEOUT_REP	RESP_TIMEOUT_REP	Reserved
			R/W	R/W	R/W	R/W	R/W	

23–19	18–16
Reserved	BURST_TIMEOUT
	R/W

15–11	10–8
Reserved	RESP_TIMEOUT
	R/W

7–5	4	3–1	0
Reserved	REJECT	Reserved	CORE_RESET
	R/W		R/W

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-6. Agent Control Register - IA Block (L3_IA_AGENT_CONTROL) Field Descriptions

Bit	Field	Value	Description
63-30	Reserved	0	Reserved
29	INBAND_ERROR_SECONDARY_REP		Reporting of in-band errors indicating debug error.
		0x0	No special reporting
		0x1	Report error
28	INBAND_ERROR_PRIMARY_REP		Reporting of in-band errors indicating application error.
		0x0	No special reporting
		0x1	Report error
27	ALL_INBAND_ERROR_REP		Reporting of all in-band errors
		0x0	Only report errors that cannot be reported in-band
		0x1	Report all in-band errors
26	BURST_TIMEOUT_REP		Open burst and ReadEx/Write timeout reporting
		0x0	No special reporting
		0x1	Report out of band
25	RESP_TIMEOUT_REP		Response timeout reporting
		0x0	No special reporting
		0x1	Report out-of-band
24-19	Reserved	0	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 9-6. Agent Control Register - IA Block (L3_IA_AGENT_CONTROL) Field Descriptions (continued)

Bit	Field	Value	Description
18-16	BURST_TIMEOUT		Response Timeout Bound:
		0x0	No timeout
		0x1	1x base cycles
		0x2	4x base cycles
		0x3	16x base cycles
		0x4	64x base cycles
15-11	Reserved	0	Reserved
10-8	RESP_TIMEOUT		Response Timeout Bound:
		0x0	No timeout
		0x1	1x base cycles
		0x2	4x base cycles
		0x3	16x base cycles
		0x4	64x base cycles
7-5	Reserved	0	Reserved
4	REJECT		Request rejection control
		0x0	Normal operation
		0x1	Block requests from the initiator
3-1	Reserved	0	Reserved
0	CORE_RESET		Reset control for agent and reset control on core
		0x0	Core reset control inactive
		0x1	Core reset control active

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.1.2.2 Agent Status Register (L3_IA_AGENT_STATUS)

The agent status register (L3_IA_AGENT_STATUS) is shown in Figure 9-2 and described in Table 9-7.

Figure 9-2. Agent Status Register (L3_IA_AGENT_STATUS)

63–48	47–32	31–30	29	28	27–24	23–17	16	15–12	11–9	8	7	6	5	4	3–1	0
Reserved	Reserved	Reserved	INBAND_ERROR_SECONDARY	INBAND_ERROR_PRIMARY	Reserved	Reserved	BURST_TIMEOUT	TIMEBASE	Reserved	RESP_TIMEOUT	READEX	BURST	RESP_WAITING	REQ_ACTIVE	Reserved	CORE_RESET
			R/W	R/W			R	R		R	R	R	R	R		R

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-7. Agent Status Register (L3_IA_AGENT_STATUS) Field Descriptions

Bit	Field	Value	Description
63-30	Reserved	0	Reserved
29	INBAND_ERROR_SECONDARY		Error Status for in-band errors indicating a debug error
		Read 0x0	No in-band error received
		Write 0x0	Ignored
		Read 0x1	In-band error received
		Write 0x1	Clear in-band error
28	INBAND_ERROR_PRIMARY		Error Status for in-band errors indicating application Error
		Read 0x0	No in-band error received
		Write 0x0	Ignored
		Read 0x1	In-band error received
		Write 0x1	Clear in-band error
27-217	Reserved	0	Reserved
16	BURST_TIMEOUT		Status of open burst and
15-12	TIMEBASE		Observation of timebase signals for internal verification
11-9	Reserved	0	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 9-7. Agent Status Register (L3_IA_AGENT_STATUS) Field Descriptions (continued)

Bit	Field	Value	Description
8	RESP_TIMEOUT		Response timeout status Reserved for instances 3, 6 to 10 and 12
7	READEX		Status of ReadEx/Write
6	BURST		Status of open burst
5	RESP_WAITING		Response Waiting Reserved for instance 3,5,7 and 8
4	REQ_ACTIVE		Requests outstanding
3-1	Reserved	0	Reserved
0	CORE_RESET		Reset input from core interface

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.1.2.3 Error Log Register - IA Block (L3_IA_ERROR_LOG)

The error log register - IA Block (L3_IA_ERROR_LOG) is shown in Figure 9-3 and described in Table 9-8.

Figure 9-3. Error Log Register - IA Block (L3_IA_ERROR_LOG)

63–48	47–32	31	30	29–28	27–24	23–16	15–8	7–3	2–0
Reserved	REQ_INFO	MULTI	SECONDARY	Reserved	CODE	Reserved	INITID	Reserved	CMD

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-8. Error Log Register - IA Block (L3_IA_ERROR_LOG) Field Descriptions

Bit	Field	Value	Description
63-48	Reserved	0	Reserved
47-32	REQ_INFO		MReqInfo bits of command that caused the error
31	MULTI		Multiple Errors
		Write 0x0	Ignored
		Read 0x0	Multiple error not seen
		Write 0x1	Clear MULTI flag
		Read 0x1	Multiple error seen
30	SECONDARY		Indicates whether error was primary or secondary
		Write 0x0	Ignored
		Read 0x0	Primary Error
		Write 0x1	Reset SECONDARY field
		Read 0x1	Secondary Error
29-28	Reserved	0	Reserved
27-24	CODE		Error code
23-16	Reserved	0	Reserved
15-8	INITID		Initiator ID from which the command was launched
7-3	Reserved	0	Reserved
2-0	CMD		Command that caused the error

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.1.2.4 Error Log Address Register - IA Block (L3_IA_ERROR_LOG_ADDR)

The error log address register - IA block (L3_IA_ERROR_LOG_ADDR) is shown in Figure 9-4 and described in Table 9-9.

Figure 9-4. Error Log Address Register - IA Block (L3_IA_ERROR_LOG_ADDR)

63 – 32	31 – 0
Reserved	ADDR
	R

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-9. Error Log Address Register - IA Block (L3_IA_ERROR_LOG_ADDR) Field Descriptions

Bit	Field	Value	Description
63-32	Reserved	0	Reserved
31-0	ADDR		Address of the command that caused the error

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.1.3 L3 Target Agent (L3 TA) Register Mapping Summary

This section describes the TA register block. Each TA in L3 interconnect has its own register block.

The following are the TA registers:

- SDRAM memory scheduler (TA_SMS module)
- General-purpose memory controller (TA_GPMC module)
- On-chip memory RAM (TA_OCM_RAM module)
- On-chip memory ROM (TA_OCM_ROM module)
- Master D2D (TA_MAD2D module)
- IVA2.2 subsystem (TA_IVA2.2 module)
- SGX subsystem (TA_SGX module)
- L4-Core interconnect (TA_L4_CORE module)
- L4-Per interconnect (TA_L4_PER module)
- L4-Emu interconnect (TA_L4_EMU module)

Table 9-10 through Table 9-13 lists all initiator target registers and their physical addresses depending on the module instance.

Figure 9-5 through Figure 9-8 describe the individual common registers in the module instance.

Table 9-10. Target Agent Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	TA_SMS Physical Address	TA_GPMC Physical Address	TA_OCM_RAM Physical Address	Section
L3_TA_AGENT_CONTROL	RW	64	0x6800 2020	0x6800 2420	0x6800 2820	Section 9.5.1.4.1
L3_TA_AGENT_STATUS	R	64	0x6800 2028	0x6800 2428	0x6800 2828	Section 9.5.1.4.2
L3_TA_ERROR_LOG	RW	64	0x6800 2058	0x6800 2458	0x6800 2858	Section 9.5.1.4.3
L3_TA_ERROR_LOG_ADDR	R	64	0x6800 2060	0x6800 2460	0x6800 2860	Section 9.5.1.4.4

Table 9-11. Target Agent Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	TA_OCM_ROM Physical Address	TA_MAD2D Physical Address	Section
L3_TA_AGENT_CONTROL	RW	64	0x6800 2C20	0x6800 3420	Section 9.5.1.4.1
L3_TA_AGENT_STATUS	R	64	0x6800 2C28	0x6800 3428	Section 9.5.1.4.2
L3_TA_ERROR_LOG	RW	64	0x6800 2C58	0x6800 3458	Section 9.5.1.4.3
L3_TA_ERROR_LOG_ADDR	R	64	0x6800 2C60	0x6800 3460	Section 9.5.1.4.4

Table 9-12. Target Agent Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	TA_IVA2.2 Physical Address	TA_SGX Physical Address	Section
L3_TA_AGENT_CONTROL	RW	64	0x6800 6020	0x6800 6420	Section 9.5.1.4.1
L3_TA_AGENT_STATUS	R	64	0x6800 6028	0x6800 6428	Section 9.5.1.4.2
L3_TA_ERROR_LOG	RW	64	0x6800 6058	0x6800 6458	Section 9.5.1.4.3
L3_TA_ERROR_LOG_ADDR	R	64	0x6800 6060	0x6800 6460	Section 9.5.1.4.4

Table 9-13. Target Agent Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	TA_L4_CORE Physical Address	TA_L4_PER Physical Address	TA_L4_EMU Physical Address	Section
L3_TA_AGENT_CONTROL	RW	64	0x6800 6820	0x6800 6C20	0x6800 7020	Section 9.5.1.4.1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 9-13. Target Agent Common Register Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	TA_L4_CORE Physical Address	TA_L4_PER Physical Address	TA_L4_EMU Physical Address	Section
L3_TA_AGENT_STATUS	RW	64	0x6800 6828	0x6800 6C28	0x6800 7028	Section 9.5.1.4.2
L3_TA_ERROR_LOG	RW	64	0x6800 6858	0x6800 6C58	0x6800 7058	Section 9.5.1.4.3
L3_TA_ERROR_LOG_ADDR	R	64	0x6800 6860	0x6800 6C60	0x6800 7060	Section 9.5.1.4.4

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.1.4 L3 Target Agent (L3 TA) Register Descriptions

9.5.1.4.1 Agent Control Register - TA Block (L3_TA_AGENT_CONTROL)

The agent control register (L3_TA_AGENT_CONTROL) is shown in Figure 9-5 and described in Table 9-14.

Figure 9-5. Agent Control Register - TA Block (L3_TA_AGENT_CONTROL)

63–48	47–32	31–26	25	24	23–16	15–11	10–8	7–5	4	3–1	0
Reserved	Reserved	Reserved	REQ	SER	Reserved	Reserved	REQ_TIMEOUT	Reserved	REJECT	Reserved	COR
			R/W	R/W			R/W		R/W		R/W

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-14. Agent Control Register - TA Block (L3_TA_AGENT_CONTROL) Field Descriptions

Bit	Field	Value	Description
63-26	Reserved	0	Reserved
25	REQ		Request Timeout Reporting
		0x0	No special reporting
		1x1	Report out of band
24	SER		SError reporting
		0x0	Suppress Serror reporting
		1x1	Report Serror
23-11	Reserved	0	Reserved
10-8	REQ_TIMEOUT		Request Timeout Bound
		0x0	No timeout
		0x1	1x base cycles
		0x2	4x base cycles
		0x3	16x base cycles
		0x4	64x base cycles
7-5	Reserved	0	Reserved
4	REJECT		Request rejection control
		0x0	Request rejection control
		1x1	Block requests to this target
3-1	Reserved	0	Reserved
0	COR		Reset output on core
		0x0	Inactive
		1x1	Reset control active

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.1.4.2 Agent Status Register (L3_TA_AGENT_STATUS)

The agent status register (L3_TA_AGENT_STATUS) is shown in Figure 9-6 and described in Table 9-15.

Figure 9-6. Agent Status Register (L3_TA_AGENT_STATUS)

63–25	24	23–17	16
Reserved	SERROR	Reserved	BURST_CLOSE
	R/W		R

15–12	11–9	8
TIMEBASE	Reserved	REQ_TIMEOUT
R		R

7	6	5	4	3–1	0
READEX	BURST	RESP_ACTIVE	REQ_WAITING	Reserved	COR_RESET
R	R	R	R		R

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-15. Agent Status Register (L3_TA_AGENT_STATUS) Field Descriptions

Bit	Field	Value	Description
63-25	Reserved	0	Reserved
24	SERROR		Serror assertion detected
			Reserved for instances 1, 3 to 6
23-17	Reserved	0	Reserved
16	BURST_CLOSE		Forced burst close status
		0x0	Normal operation
		0x1	Burst close command
15-12	TIMEBASE		Observation of timebase signals
11-9	Reserved	0	Reserved
8	REQ_TIMEOUT		Request timeout status
		0x0	Normal operation
		0x1	Request timed out, responding ERR to all the requests
7	READEX		Status of readEx/Write
		0x0	No pending ReadEx
		0x1	ReadEx pending on at lease one thread
6	BURST		Status of open burst
		0x0	No open burst
		0x1	Open burst on at least one thread
5	RESP_ACTIVE		Responses outstanding
		0x0	No responses outstanding
		0x1	Response outstanding in the target
4	REQ_WAITING		Requests waiting
		0x0	No request waiting
		0x1	Request waiting for acceptance by target
3-1	Reserved	0	Reserved
0	CORE_RESET		Reset input from core interface
		0x0	Reset inactive
		0x1	Reset active

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.1.4.3 Error Log Register - TA Block (L3_TA_ERROR_LOG)

This register logs the error detected by a target agent. The error log register (L3_TA_ERROR_LOG) is shown in Figure 9-7 and described in Table 9-16.

Figure 9-7. Error Log Register - TA Block (L3_TA_ERROR_LOG)

63–48	47–42	41–32
Reserved	Reserved	REQ_INFO
		R

31	30–28	27–24
MULTI	Reserved	CODE
R/W		R/W

23–16
Reserved

15–8
INITID
R

7–3	2–0
Reserved	CMD
	R

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-16. Error Log Register - TA Block (L3_TA_ERROR_LOG) Field Descriptions

Bit	Field	Value	Description
63-42	Reserved	0	Reserved
41-32	REQ_INFO		MReqInfo bits of command that caused the error
31	MULTI		Multiple Errors
		Write 0x0:	Ignored
		Read 0x0:	Multiple error not seen
		Write 0x1:	Clear MULTI flag
		Read 0x1:	Multiple error seen
30-28	Reserved	0	Reserved
27-24	CODE		Error code
23-16	Reserved	0	Reserved
15-8	INITID		Initiator ID from which command was launched
7-3	Reserved	0	Reserved
2-0	CMD		Command that caused the error

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.1.4.4 Error Log Address Register - TA Block (L3_TA_ERROR_LOG_ADDR)

The error log address register - TA block (L3_TA_ERROR_LOG_ADDR) is shown in Figure 9-8 and described in Table 9-17.

Figure 9-8. Error Log Address Register - TA Block (L3_TA_ERROR_LOG_ADDR)

63 – 40	31 – 0
Reserved	ADDR
	R

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-17. Error Log Address Register - TA Block (L3_TA_ERROR_LOG_ADDR) Field Descriptions

Bit	Field	Value	Description
63-40	Reserved	0	Reserved
31-0	ADDR		Address of the command that caused the error

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.1.5 Register Target (RT) Register Mapping Summary

This section describes the RT module.

Table 9-18 lists the RT registers and their physical addresses.

Figure 9-9 through Figure 9-11 describe the individual registers in the module instance.

Table 9-18. RT Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address	Section
L3_RT_NETWORK	R	64	0x010	0x6800 0010	Section 9.5.1.6.1
L3_RT_INITID_READBACK	R	64	0x070	0x6800 0070	Section 9.5.1.6.2
L3_RT_NETWORK_CONTROL	RW	64	0x078	0x6800 0078	Section 9.5.1.6.3

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.1.6 Register Target (RT) Register Descriptions

9.5.1.6.1 L3_RT_NETWORK

This register identifies the interconnect and is present only in the register target. The L3_RT_NETWORK is shown in Figure 9-9 and described in Table 9-19.

Figure 9-9. L3_RT_NETWORK

63–32	31–0
ID	Reserved
R	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-19. L3_RT_NETWORK Field Descriptions

Bit	Field	Value	Description
63-32	ID		Unique Interconnect ID
31-0	Reserved	0	Reserved

9.5.1.6.2 L3_RT_INITID_READBACK

This register is used by initiators to discover their own identity. The L3_RT_INITID_READBACK is shown in Figure 9-10 and described in Table 9-20.

Figure 9-10. L3_RT_INITID_READBACK

31–8	7–0
Reserved	INITID
	R

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-20. L3_RT_INITID_READBACK Field Descriptions

Bit	Field	Value	Description
63-8	Reserved	0	Reserved
7-0	INITID		Returns initiator ID of core thread that initiated the read

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.1.6.3 L3_RT_NETWORK_CONTROL

This register controls such interconnect wide functions as the timeout base scale and the disabling of fine grained hardware clock gating. The L3_RT_NETWORK_CONTROL is shown in Figure 9-11 and described in Table 9-21.

Figure 9-11. L3_RT_NETWORK_CONTROL

63–57	56	55–32
Reserved	CLO	Reserved
	R/W	

31–25	24	23–11	10–8	7–0
Reserved			TIMEOUT_BASE	Reserved
			R/W	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-21. L3_RT_NETWORK_CONTROL Field Descriptions

Bit	Field	Value	Description
63-57	Reserved	0	Reserved
56	CLOCK_GATE_DISABLE		Overrides fine grained hardware clock gating
		0	
		1	
55-11	Reserved	0	Reserved
10-8	TIMEOUT_BASE		Timeout base period in register target clock cycles
			Program the timeout base period. Each of the agent timeout features is programmed as a multiple of the timeout base period. These timeout bases are:
		0x0	Timeout disabled
		0x1	L3 interconnect clock cycles divided by 64
		0x2	L3 interconnect clock cycles divided by 256
		0x3	L3 interconnect clock cycles divided by 1024
		0x4	L3 interconnect clock cycles divided by 4096
7-0	Reserved	0	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.1.7 Protection Mechanism (PM) Register Mapping Summary

This section describes the protection mechanism register block.

The following are the protection mechanism registers:

- Register target (PM_RT module)
- General-purpose memory controller (PM_GPMC module)
- On-chip RAM (PM_OCM_RAM module)
- On-chip ROM (PM_OCM_ROM module)
- IVA2.2 subsystem (PM_IVA2.2 module)

Table 9-22 and Table 9-24 list the protection registers and their physical addresses, depending on the module instance.

Figure 9-12 through Table 9-30 describe the individual common registers in the module instance.

Table 9-22. Protection Mechanism Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	PM_RT Physical Address	PM_GPMC Physical Address	Section
L3_PM_ERROR_LOG	RW	64	0x6801 0020	0x6801 2420	Section 9.5.1.8.1
L3_PM_CONTROL	RW	64	0x6801 0028	0x6801 2428	Section 9.5.1.8.2
L3_PM_ERROR_CLEAR_SINGLE	R	64	0x6801 0030	0x6801 2430	Section 9.5.1.8.3
L3_PM_ERROR_CLEAR_MULTI	R	64	0x6801 0038	0x6801 2438	Section 9.5.1.8.4
L3_PM_REQ_INFO_PERMISSION_i[(1)]	RW	64	0x6801 0048 + (0x20*i)	0x6801 2448 + (0x20*i)	Section 9.5.1.8.5
L3_PM_READ_PERMISSION_i[(1)]	RW	64	0x6801 0050 + (0x20*i)	0x6801 2450 + (0x20*i)	Section 9.5.1.8.6
L3_PM_WRITE_PERMISSION_i[(1)]	RW	64	0x6801 0058 + (0x20*i)	0x6801 2458 + (0x20*i)	Section 9.5.1.8.7
L3_PM_ADDR_MATCH_k[(2)]	RW	64	0x6801 0060 + (0x20*k)	0x6801 2460 + (0x20*k)	Section 9.5.1.8.9

(1) i = 0 to 1 for PM_RT
i = 0 to 7 for PM_GPMC

(2) k = 1 to 1 for PM_RT
k = 1 to 7 for PM_GPMC

Table 9-23. Protection Mechanism Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	PM_OCM_RAM Physical Address	PM_OCM_ROM Physical Address	Section
L3_PM_ERROR_LOG	RW	64	0x6801 2820	0x6801 2C20	Section 9.5.1.8.1
L3_PM_CONTROL	RW	64	0x6801 2828	0x6801 2C28	Section 9.5.1.8.2
L3_PM_ERROR_CLEAR_SINGLE	R	64	0x6801 2830	0x6801 2C30	Section 9.5.1.8.3
L3_PM_ERROR_CLEAR_MULTI	R	64	0x6801 2838	0x6801 2C38	Section 9.5.1.8.4
L3_PM_REQ_INFO_PERMISSION_i[(1)]	RW	64	0x6801 2848 + (0x20*i)	0x6801 2C48 + (0x20*i)	Section 9.5.1.8.5
L3_PM_READ_PERMISSION_i[(1)]	RW	64	0x6801 2850 + (0x20*i)	0x6801 2C50 + (0x20*i)	Section 9.5.1.8.6

(1) i = 0 to 1 for PM_OCM_ROM
i = 0 to 7 for PM_OCM_RAM

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 9-23. Protection Mechanism Common Register Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	PM_OCM_RAM Physical Address	PM_OCM_ROM Physical Address	Section
L3_PM_WRITE_PERMISSION_i[(1)]	RW	64	0x6801 2858 + (0x20*i)	0x6801 2C58 + (0x20*i)	Section 9.5.1.8.7
L3_PM_ADDR_MATCH_k[(2)]	RW	64	0x6801 2860 + (0x20*k)	0x6801 2C60 + (0x20*k)	Section 9.5.1.8.9

(2) k = 1 to 1 for PM_OCM_ROM
k = 1 to 7 for PM_OCM_RAM

Table 9-24. Protection Mechanism Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	PM_MAD2D Physical Address	PM_IVA2.2 Physical Address	Section
L3_PM_ERROR_LOG	RW	64	0x6801 3020	0x6801 4020	Section 9.5.1.8.1
L3_PM_CONTROL	RW	64	0x6801 3028	0x6801 4028	Section 9.5.1.8.2
L3_PM_ERROR_CLEAR_SINGLE	R	64	0x6801 3030	0x6801 4030	Section 9.5.1.8.3
L3_PM_ERROR_CLEAR_MULTI	R	64	0x6801 3038	0x6801 4038	Section 9.5.1.8.4
L3_PM_REQ_INFO_PERMISSION_i[(1)]	RW	64	0x6801 3048 + (0x20*i)	0x6801 4048 + (0x20*i)	Section 9.5.1.8.5
L3_PM_READ_PERMISSION_i[(1)]	RW	64	0x6801 3050 + (0x20*i)	0x6801 4050 + (0x20*i)	Section 9.5.1.8.6
L3_PM_WRITE_PERMISSION_i[(1)]	RW	64	0x6801 3058 + (0x20*i)	0x6801 4058 + (0x20*i)	Section 9.5.1.8.7
L3_PM_ADDR_MATCH_k[(2)]	RW	64	0x6801 3060 + (0x20*k)	0x6801 4060 + (0x20*k)	Section 9.5.1.8.9

(1) i = 0 to 7 for PM_MAD2D
i = 0 to 3 for PM_IVA2.2
(2) k = 1 to 7 for PM_MAD2D
k = 1 to 3 for PM_IVA2.2

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.1.8 Protection Mechanism (PM) Register Descriptions

9.5.1.8.1 L3_PM_ERROR_LOG

This register logs errors detected by the protection mechanism. The L3_PM_ERROR_LOG is shown in Figure 9-12 and described in Table 9-25.

Figure 9-12. L3_PM_ERROR_LOG

63–48	47–32	31	30	29–28	27–24	23–21	20–16	15–8	7	6–4	3	2–0
Reserved	Reserved	MULTI	SECONDARY	Reserved	CODE	Reserved	REQ_INFO	INITID	Reserved	REGION	Reserved	CMD
		RW1toClr	RW1toClr		RW1toClr		R	R		R		R

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-25. L3_PM_ERROR_LOG Field Descriptions

Bit	Field	Value	Description
63-32	Reserved	0	Reserved
31	MULTI		Multiple errors
		0x0	Multiple errors not seen
		0x1	Multiple errors seen
30	SECONDARY		Secondary error present
29-28	Reserved	0	Reserved
27-24	CODE		Error Code
23-21	Reserved	0	Reserved
20-16	REQ_INFO		MReqInfo bits of command selected for protection checking
16-8	INITID		Initiator ID from which the command was launched
7	Reserved	0	Reserved
6-4	REGION		Protection region number that command mapped to
3	Reserved	0	Reserved
2-0	CMD		Command that caused the error

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.1.8.2 L3_PM_CONTROL

This register controls protection mechanism functions such as error reporting. The L3_PM_CONTROL is shown in Figure 9-13 and described in Table 9-26.

Figure 9-13. L3_PM_CONTROL

63–16	15–26	25	24
Reserved	Reserved	ERROR_SECONDARY_REP	ERROR_REP
		R/W	R/W
			0

23–0
Reserved

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-26. L3_PM_CONTROL Field Descriptions

Bit	Field	Value	Description
63-26	Reserved	0	Reserved
25	ERROR_SECONDARY_REP		Out of band error reporting
		0x0	Out of band error reporting suppress
		0x1	Out of band error report
24	ERROR_REP		Out of band error reporting
		0x0	Out of band error reporting suppress
		0x1	Out of band error report
23-0	Reserved	0	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.1.8.3 L3_PM_ERROR_CLEAR_SINGLE

This register is read to clear single errors from the error log. The L3_PM_ERROR_CLEAR_SINGLE is shown in Figure 9-14 and described in Table 9-27.

Figure 9-14. L3_PM_ERROR_CLEAR_SINGLE

63	32
Reserved	

31	1	0
Reserved		CLEAR
		R

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-27. L3_PM_ERROR_CLEAR_SINGLE Field Descriptions

Bit	Field	Value	Description
63-1	Reserved	0	Reserved
0	CLEAR		Clear single error from log

9.5.1.8.4 L3_PM_ERROR_CLEAR_MULTI

This register is read to clear multiple errors from the error log. The L3_PM_ERROR_CLEAR_MULTI is shown in Figure 9-15 and described in Table 9-28.

Figure 9-15. L3_PM_ERROR_CLEAR_MULTI

63	32
Reserved	

31	1	0
Reserved		CLEAR
		R

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-28. L3_PM_ERROR_CLEAR_MULTI Field Descriptions

Bit	Field	Value	Description
63-1	Reserved	0	Reserved
0	CLEAR		Clear multiped error from log

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.1.8.5 L3_PM_REQ_INFO_PERMISSION_i

This register configures a protection region's permissions using the MReqInfo bits selected for the PM by the structural configuration. The L3_PM_REQ_INFO_PERMISSION_i is shown in Figure 9-16 and described in Table 9-29.

Figure 9-16. L3_PM_REQ_INFO_PERMISSION_i

63	32
Reserved	

31	16	15	0
Reserved		REQ_INFO	
		R/W	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-29. L3_PM_REQ_INFO_PERMISSION_i Field Descriptions

Bit	Field	Value	Description
63-1	Reserved	0	Reserved
0	REQ_INFO		Request info permission bits for region 0

Table 9-30. Reset Value for REQ_INFO_PERMISSION

	Regions							
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
PM_RT	0xFFFF	0x000						
PM_GPMC	0x0000	0x----	0x----	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF
PM_OCM_RAM	0x0000	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF
PM_OCM_ROM	0x----	0xFFFF						
PM_MAD2D	0x0000	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF
PM_IVA2.2	0x0000	0xFFFF	0xFFFF	0xFFFF				

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.1.8.6 L3_PM_READ_PERMISSION_i

This register configures protection region permissions for read incoming commands. The L3_PM_READ_PERMISSION_i is shown in Figure 9-17 and described in Table 9-31.

Figure 9-17. L3_PM_READ_PERMISSION_i

63							16
Reserved							

15	14	13	12	11	10	9	8
Reserved	SGX	Reserved	DAP	CAM	IVA2_MMU	USB_HS_Host	DISP_SS
	R/W		R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
Reserved		SAD2D	USB_HS_OTG	SDMA	IVA2_DMA	MPU	Reserved
		R/W	R/W	R/W	R/W	R/W	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-31. L3_PM_READ_PERMISSION_i Field Descriptions

Bit	Field	Value	Description
63-15	Reserved	0	Reserved
14	SGX		Read permission for the SGX
13	Reserved	0	Reserved
12	DAP		Read permission for the DAP
11	CAM		Read permission for the CAMERA SS
10	IVA2_MMU		Read permission for the IVA2 MMU
9	USB_HS_Host		Read permission for the USB_HS_Host
8	DISP_SS		Read permission for the DISPLAY SS
7-6	Reserved	0	Reserved
5	SAD2D		Read permission for the SAD2D
4	USB_HS_OTG		Read permission for the USB_HS_OTG
3	SDMA		Read permission for the system DMA
2	IVA2_DMA		Read permission for the IVA2
1	MPU		Read permission for the MPU
0	Reserved	0	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.1.8.7 L3_PM_WRITE_PERMISSION_i

This register configures protection region permissions for write incoming commands. The L3_PM_WRITE_PERMISSION_i is shown in Figure 9-18 and described in Table 9-32.

Figure 9-18. L3_PM_WRITE_PERMISSION_i

63							16
Reserved							

15	14	13	12	11	10	9	8
Reserved	SGX	Reserved	DAP	CAM	IVA2_MMU	USB_HS_Host	DISP_SS
	R/W		R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
Reserved		SAD2D	USB_HS_OTG	SDMA	IVA2_DMA	MPU	Reserved
		R/W	R/W	R/W	R/W	R/W	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-32. L3_PM_WRITE_PERMISSION_i Field Descriptions

Bit	Field	Value	Description
63-15	Reserved	0	Reserved
14	SGX		Write permission for the SGX
13	Reserved	0	Reserved
12	DAP		Write permission for the DAP
11	CAM		Write permission for the CAMERA SS
10	IVA2_MMU		Write permission for the IVA2 MMU
9	USB_HS_Host		Write permission for the USB_HS_Host
8	DISP_SS		Write permission for the DISPLAY SS
7-6	Reserved	0	Reserved
5	SAD2D		Write permission for the SAD2D
4	USB_HS_OTG		Write permission for the USB_HS_OTG
3	SDMA		Write permission for the system DMA
2	IVA2_DMA		Write permission for the IVA2
1	MPU		Write permission for the MPU
0	Reserved	0	Reserved

9.5.1.8.8 Bit Availability and Initialization Values for L3_PM_READ_PERMISSION_i and L3_PM_WRITE_PERMISSION_i

Table 9-33 shows bit available in L3_PM_READ_PERMISSION_i and L3_PM_WRITE_PERMISSION_i registers. All N/A are considered as reserved bits with a Read access.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 9-33. Bit Availability and Initialization Values for L3_PM_READ_PERMISSION_i and L3_PM_WRITE_PERMISSION_i[(1)]

PM	BITS											
	1: MPU	2: IVA2_ DMA	3: SDMA	4: USB_HS_ OTG	5: SAD2D	8: DISP SS	9: USB_HS_ Host	10: IVA2_ MMU	11: CAM SS	12: DAP	14: SGX	63-15: Reserved
PM_RT region 0 to 1	1	1	N/A	N/A	N/A	N/A	N/A	1	N/A	1	N/A	0
PM_GPMC region 0 to 7	1	1	1	1	1	N/A	1	1	N/A	1	1	1
PM_OCM_RAM region 0 to 7	1	1	1	1	1	1	1	1	1	1	1	1
PM_OCM_ROM region 0 to 1 (ro)[(2)]	1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	1	N/A	0
PM_MAD2D region 0 to 7	1	1	1	1	N/A	1	1	1	1	1	1	1
PM_IVA2.2 region 0 to 3	1	1	1	N/A	N/A	N/A	N/A	1	N/A	1	N/A	0

(1) Some firewall reset value are exported from the system control module. Values in this table are valid for GP device only. HS value may differ.

(2) ROM is a read only memory; therefore, the write permission is set to 0. The value for Region 0 is exported from control module (PLATFORM_MPU_OCMROM_DT_FW_RD).

9.5.1.8.9 L3_PM_ADDR_MATCH_k

The L3_PM_ADDR_MATCH_k is shown in Figure 9-19 and described in Table 9-34.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 9-19. L3_PM_ADDR_MATCH_k

63–20	19–10	9	8	7–3	2–0
Reserved	BASE_ADDR	LEVEL	Reserved	SIZE	ADDR_SPACE
	R	R		R	R

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-34. L3_PM_ADDR_MATCH_k Field Descriptions

Bit	Field	Value	Description
63-20	Reserved	0	Reserved
19-10	BASE_ADDR		Protection region base address
9	LEVEL		Protection region level
8	Reserved	0	Reserved
7-3	SIZE		Protection region size
2-0	ADDR_SPACE		Protection region address space

Table 9-35. Reset Value for L3_PM_ADDR_MATCH_k[(1)]

PM	Region	BASE ADDRESS	LEVEL	SIZE	ADDR_SPACE
PM_RT	1	0x040	0x1	0x06	0x0
PM_GPMC	1	0x000	0x0	0x23	0x0
	2	0x000	0x0	0x00	0x0
	3	0x000	0x0	0x00	0x0
	4	0x000	0x0	0x00	0x0
	5	0x000	0x0	0x00	0x0
	6	0x000	0x0	0x00	0x0
	7	0x000	0x0	0x00	0x0
PM_OCM_RAM	1	0x000	0x0	0x00	0x0
	2	0x03E	0x0	0x02	0x0
	3	0x000	0x0	0x00	0x0
	4	0x000	0x0	0x00	0x0
	5	0x000	0x0	0x00	0x0
	6	0x000	0x0	0x00	0x0
	7	0x000	0x0	0x00	0x0
PM_OCM_ROM	1	0x050	0x0	0x05	0x0
	2	0x000	0x0	0x00	0x0
	3	0x000	0x0	0x00	0x0
PM_MAD2D	1	0x000	0x0	0x00	0x0
	2	0x000	0x0	0x00	0x0
	3	0x000	0x0	0x00	0x0
	4	0x000	0x0	0x00	0x0
	5	0x000	0x0	0x00	0x0
	6	0x000	0x0	0x00	0x0
	7	0x000	0x0	0x00	0x0

(1) Some firewall reset value are exported from the system control module. Values in this table are valid for GP device only. HS value may differ.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 9-35. Reset Value for L3_PM_ADDR_MATCH_k[(1)] (continued)

PM	Region	BASE ADDRESS	LEVEL	SIZE	ADDR_SPACE
PM_IVA2	1	0x000	0x0	0x00	0x0
	2	0x000	0x0	0x00	0x0
	3	0x000	0x0	0x00	0x0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.1.9 Sideband Interconnect (SI) Register Mapping Summary

This section describes the sideband interconnect register block.

Table 9-36 lists the SI registers and their physical addresses.

Figure 9-20 through Figure 9-22 describe the individual registers in the module instance.

Table 9-36. SI Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address	Section
L3_SI_CONTROL	RW	64	0x020	0x6800 0420	Section 9.5.1.10.1
L3_SI_FLAG_STATUS_0	R	64	0x110	0x6800 0510	Section 9.5.1.10.2
L3_SI_FLAG_STATUS_1	R	64	0x130	0x6800 0530	Section 9.5.1.10.3

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.1.10 Sideband Interconnect (SI) Register Descriptions

9.5.1.10.1 L3_SI_CONTROL

Control of register and sideband interconnect. The L3_SI_CONTROL is shown in Figure 9-20 and described in Table 9-37.

Figure 9-20. L3_SI_CONTROL

63	57	56
Reserved		CLOCK_GATE_DISABLE
		R/W
		0

55
Reserved

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-37. L3_SI_CONTROL Field Descriptions

Bit	Field	Value	Description
63-57	Reserved	0	Reserved
56	CLOCK_GATE_DISABLE		Overrides fine grained hardware clock gating in register and sideband interconnect
		0x0	Normal clock gating
		0x1	Clock gating disabled
55-0	Reserved	0	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.1.10.2 L3_SI_FLAG_STATUS_0

This register is used to observe the individual bits that make up a composite interconnect flag. The L3_SI_FLAG_STATUS_0 is shown in Figure 9-21 and described in Table 9-38.

Figure 9-21. L3_SI_FLAG_STATUS_0

63	0
STATUS	
R	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-38. L3_SI_FLAG_STATUS_0 Field Descriptions

Bit	Field	Value	Description
63-0	STATUS		Status of sideband signals making up composite interconnect flag for application.

9.5.1.10.3 L3_SI_FLAG_STATUS_1

This register is used to observe the individual bits that make up a composite interconnect flag. The L3_SI_FLAG_STATUS_1 is shown in Figure 9-22 and described in Table 9-39.

Figure 9-22. L3_SI_FLAG_STATUS_1

63	0
STATUS	
R	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-39. L3_SI_FLAG_STATUS_1 Field Descriptions

Bit	Field	Value	Description
63-0	STATUS		Status of sideband signals making up composite interconnect flag for debug.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.2 L4 Interconnects Registers

A summary of the hardware interface for the L4-Core, L4-Per, L4-Emu and L4-Wkup interconnects is given in Table 9-40 to Table 9-43. Each module instance in the design is shown with the module register map and bit definitions for each bit field.

Table 9-40. L4- Core Instance Summary

Module Name	Base Address	Size
CORE_TA_CONTROL	0x4800 3000	4K bytes
CORE_TA_CM	0x4800 7000	4K bytes
CORE_AP	0x4804 0000	2K bytes
CORE_IA	0x4804 0800	2K bytes
CORE_LA	0x4804 1000	4K bytes
CORE_TA_DISPLAY_SS	0x4805 1000	4K bytes
CORE_TA_SDMA	0x4805 7000	4K bytes
CORE_TA_I2C3	0x4806 1000	4K bytes
CORE_TA_USB_HS_TLL	0x4806 3000	4K bytes
CORE_TA_USB_HS_Host	0x4806 5000	4K bytes
CORE_TA_UART1	0x4806 B000	4K bytes
CORE_TA_UART2	0x4806 D000	4K bytes
CORE_TA_I2C1	0x4807 1000	4K bytes
CORE_TA_I2C2	0x4807 3000	4K bytes
CORE_TA_MCBSP1	0x4807 5000	4K bytes
CORE_TA_GPTIMER10	0x4808 7000	4K bytes
CORE_TA_GPTIMER11	0x4808 9000	4K bytes
CORE_TA_MAILBOX	0x4809 5000	4K bytes
CORE_TA_MCBSP5	0x4809 7000	4K bytes
CORE_TA_SPI1	0x4809 9000	4K bytes
CORE_TA_SPI2	0x4809 B000	4K bytes
CORE_TA_MMCHS1	0x4809 D000	4K bytes
CORE_TA_MSPRO	0x4809 F000	4K bytes
CORE_TA_RNG1	0x480A 1000	4K bytes
CORE_TA_D3D1	0x480A 3000	4K bytes
CORE_TA_SHAM2	0x480A 5000	4K bytes
CORE_TA_AES1	0x480A 7000	4K bytes
CORE_TA_FPKA	0x480A A000	4K bytes
CORE_TA_USB_HS_OTG	0x480A C000	4K bytes
CORE_TA_MMCHS3	0x480A E000	4K bytes
CORE_TA_MG	0x480B 1000	4K bytes
CORE_TA_HD1W	0x480B 3000	4K bytes
CORE_TA_MMCHS2	0x480B 5000	4K bytes
CORE_TA_ICR	0x480B 7000	4K bytes
CORE_TA_SPI3	0x480B 9000	4K bytes
CORE_TA_SPI4	0x480B B000	4K bytes
CORE_TA_CAMERA	0x480C 0000	4K bytes
CORE_TA_DES2	0x480C 2000	4K bytes
CORE_TA_SHAM1	0x480C 4000	4K bytes
CORE_TA_AES2	0x480C 6000	4K bytes
CORE_TA_INTH	0x480C 8000	4K bytes
CORE_TA_ICR	0x480C E000	4K bytes
CORE_TA_WKUP	0x4834 0000	4K bytes

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 9-41. L4-Per Instance Summary

Module Name	Base Address	Size
PER_AP	0x4900 0000	2K bytes
PER_IA	0x4900 0800	2K bytes
PER_LA	0x4900 1000	4K bytes
PER_TA_UART3	0x4902 1000	512 bytes
PER_TA_MCBSP2	0x4902 3000	1K byte
PER_TA_MCBSP3	0x4902 5000	1K byte
PER_TA_MCBSP4	0x4902 7000	1K byte
PER_TA_MCBSP_SIDETONE2	0x4902 9000	4K bytes
PER_TA_MCBSP_SIDETONE3	0x4902 B000	4K bytes
PER_TA_WDtimer3	0x4903 1000	2K bytes
PER_TA_GPTIMER2	0x4903 3000	1K byte
PER_TA_GPTIMER3	0x4903 5000	1K byte
PER_TA_GPTIMER4	0x4903 7000	1K byte
PER_TA_GPTIMER5	0x4903 9000	1K byte
PER_TA_GPTIMER6	0x4903 B000	1K byte
PER_TA_GPTIMER7	0x4903 D000	1K byte
PER_TA_GPTIMER8	0x4903 F000	1K byte
PER_TA_GPTIMER9	0x4904 1000	1K byte
PER_TA_GPIO2	0x4905 1000	1K byte
PER_TA_GPIO3	0x4905 3000	1K byte
PER_TA_GPIO4	0x4905 5000	1K byte
PER_TA_GPIO5	0x4905 7000	1K byte
PER_TA_GPIO6	0x4905 9000	1K byte

Table 9-42. L4-Emu Instance Summary

Module Name	Base Address	Size
EMU_TEST_TAP	0x5400 5000	4K bytes
EMU_AP	0x5400 6000	2K bytes
EMU_IA_0	0x5400 6800	2K bytes
EMU_LA	0x5400 7000	4K bytes
EMU_IA_1	0x5400 8000	2K bytes
EMU_TA_MPU	0x5401 8000	4K bytes
EMU_TA_TPIU	0x5401 A000	4K bytes
EMU_TA_ETB	0x5401 C000	4K bytes
EMU_TA_DAPCTL	0x5401 E000	4K bytes
EMU_TA_SDTI	0x5401 F000	4K bytes
EMU_TA_L4WKUP	0x5473 0000	4K bytes

Table 9-43. L4-WKUP Instance Summary

Module Name	Base Address	Size
WKUP_TA_GPTIMER12	0x4830 5000	4K bytes
WKUP_TA_PRM	0x4830 9000	4K bytes
WKUP_TA_WDTIMER1	0x4830 D000	4K bytes
WKUP_TA_USIM	0x4830 F000	4K bytes
WKUP_TA_GPIO1	0x4831 1000	4K bytes
WKUP_TA_WDTIMER2	0x4831 5000	4K bytes

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 9-43. L4-WKUP Instance Summary (continued)

Module Name	Base Address	Size
WKUP_TA_GPTIMER1	0x4831 9000	4K bytes
WKUP_TA_SYNCTIMER32K	0x4832 1000	4K bytes
WKUP_AP	0x4832 8000	2K bytes
WKUP_IA_L4CORE	0x4832 8800	2K bytes
WKUP_LA	0x4832 9000	4K bytes
WKUP_IA_L4EMU	0x4832 A000	2K bytes

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.2.1 L4 Initiator Agent (L4 IA) Register Mapping Summary

This section provides information on the L4 IA module. Each of the registers within the module instance is described separately in Table 9-44 to Table 9-45.

The initiator OCP interface register block (IA) consists of the status, control, and error log registers that can be used to configure the interface of an initiator OCP. There can be only one register block for each initiator OCP interface.

Table 9-44. L4 IA Register Mapping Summary (1)

Register Name	Type	Register Width (Bits)	CORE_IA Physical Address	PER_IA Physical Address	EMU_IA_ L3 Physical Address	Section
L4_IA_AGENT_CONTROL_L	RW	32	0x4804 0820	0x4900 0820	0x5400 6820	Section 9.5.2.2.1
L4_IA_AGENT_STATUS_L	RW	32	0x4804 0828	0x4900 0828	0x5400 6828	Section 9.5.2.2.2
L4_IA_ERROR_LOG_L	RW	32	0x4804 0858	0x4900 0858	0x5400 6858	Section 9.5.2.2.3

Table 9-45. L4 IA Register Mapping Summary (2)

Register Name	Type	Register Width (Bits)	EMU_IA_DAP Physical Address	WKUP_IA_EMU Physical Address	WKUP_IA_CORE Physical Address	Section
L4_IA_AGENT_CONTROL_L	RW	32	0x5400 8020	0x4832 8820	0x4832 A020	Section 9.5.2.2.1
L4_IA_AGENT_STATUS_L	RW	32	0x5400 8028	0x4832 8828	0x4832 A028	Section 9.5.2.2.2
L4_IA_ERROR_LOG_L	RW	32	0x5400 8058	0x4832 8858	0x4832 A058	Section 9.5.2.2.3

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.2.2 L4 Initiator Agent (L4 IA) Register Descriptions

9.5.2.2.1 L4_IA_AGENT_CONTROL_L

This register enables error reporting on an initiator interface. The error reporting mechanism is enabled when the INBAND_ERROR_REP bit field is set to 1. The out-of-band OCP MError reporting mechanism is enabled when the MERROR_REP bit field is set to 1.

The L4_IA_AGENT_CONTROL_L is shown in Figure 9-23 and described in Table 9-46.

Figure 9-23. L4_IA_AGENT_CONTROL_L

31 28	27	26 25	24
Reserved	INBAND_ ERROR_REP	Reserved	MERROR_REP
	R		R

23 0
Rserved

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-46. L4_IA_AGENT_CONTROL_L Field Descriptions

Bit	Field	Value	Description
3128	Reserved	0	Read returns 0
27	INBAND_ ERROR_REP		Setting this field to 1 reports on in-band errors using the INBAND_ERROR log bit of IA.AGENT_STATUS register.
26-25	Reserved	0	Read returns 0
24	MERROR_REP		Enable MError reporting
23-0	Reserved	0	Read returns 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.2.2.2 L4_IA_AGENT_STATUS_L

This register stores status information for an initiator. The INBAND_ERROR and MERROR fields are read/write and are implemented as log bits.

The L4_IA_AGENT_STATUS_L is shown in Figure 9-24 and described in Table 9-47.

Figure 9-24. L4_IA_AGENT_STATUS_L

31–28	27	26–25	24
Reserved	INBAND_ ERROR_REP	Reserved	MERROR_REP
	R 1toClr		R

23–0
Rserved
0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-47. L4_IA_AGENT_STATUS_L Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Reserved
27	INBAND_ ERROR_REP	0x0 0x1	No In-Band error present In-Band error present
26-25	Reserved	0	Reserved
24	MERROR_REP	0x0 0x1	No MError error present MErorr error present
23-0	Reserved	0	Reserved

9.5.2.2.3 L4_IA_ERROR_LOG_L

This register logs information about error conditions. The CODE field logs any protection violation or address hole errors detected by the initiator subsystem while decoding a request.

The L4_IA_ERROR_LOG_L is shown in Figure 9-25 and described in Table 9-48.

Figure 9-25. L4_IA_ERROR_LOG_L

31	30–26	25–24
MULTI	Reserved	CODE
R/W		R/W

23–0
Reserved
0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-48. L4_IA_ERROR_LOG_L Field Descriptions

Bit	Field	Value	Description
31	MULTI		While the CODE field is not zero, the MULTI bit is asserted whenever an additional error is detected. Once set by hardware, the MULTI bit can only be cleared by writing a 1 to it while writing a value other than zero to the CODE field.
30-26	Reserved	0	Read returns 0.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 9-48. L4_IA_ERROR_LOG_L Field Descriptions (continued)

Bit	Field	Value	Description
25-24	CODE		The error code of an initiator request.
		0x00	No errors
		0x01	Reserved
		0x10	Address hole
		0x11	Protection violation
			The CODE field, once set by hardware, can only be cleared by writing a non-zero value to it, in conjunction with writing a 1 to the MULTI bit field.
23-0	Reserved	0	Read returns 0.

9.5.2.3 L4 Target Agent (L4 TA) Register Mapping Summary

This section provides information on the L4 Target agent (TA) register module. Each of the registers within the module instance is described separately in Table 9-49 to Table 9-77.

Table 9-49 to Table 9-78 provide information on the L4 Target Agent having two supplementary registers OCP_CONTROL and OCP_STATUS

The TA register block consists of the status and control registers that can be used to configure the interfaces of the targets.

Table 9-49. CORE_TA Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	CORE_TA_CONTROL Physical Address	Section
L4_TA_AGENT_CONTROL_L	RW	32	0x4800 3020	Section 9.5.2.4.1
L4_TA_AGENT_CONTROL_H	RW	32	0x4800 3024	Section 9.5.2.4.2
L4_TA_AGENT_STATUS_L	RW	32	0x4800 3028	Section 9.5.2.4.3

Table 9-50. CORE_TA Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	CORE_TA_CM Physical Address	CORE_TA_ DISPLAY_SS Physical Address	CORE_ TA_SDMA Physical Address	Section
L4_TA_AGENT_CONTROL_L	RW	32	0x4802 7020	0x4805 1020	0x4805 7020	Section 9.5.2.4.1
L4_TA_AGENT_CONTROL_H	RW	32	0x4802 7024	0x4805 1024	0x4805 7024	Section 9.5.2.4.2
L4_TA_AGENT_STATUS_L	RW	32	0x4802 7028	0x4805 1028	0x4805 7028	Section 9.5.2.4.3

Table 9-51. CORE_TA Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	CORE_TA_I2C3 Physical Address	Section
L4_TA_AGENT_CONTROL_L	RW	32	0x4806 1020	Section 9.5.2.4.1
L4_TA_AGENT_CONTROL_H	RW	32	0x4806 1024	Section 9.5.2.4.2
L4_TA_AGENT_STATUS_L	RW	32	0x4806 1028	Section 9.5.2.4.3

Table 9-52. CORE_TA Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	CORE_TA_ USB_TLL Physical Address	CORE_TA_ USB_HS_Host Physical Address	CORE_ TA_UART1 Physical Address	Section
L4_TA_AGENT_CONTROL_L	RW	32	0x4806 3020	0x4806 5020	0x4806 B020	Section 9.5.2.4.1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 9-52. CORE_TA Common Register Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	CORE_TA_ USB_TLL Physical Address	CORE_TA_ USB_HS_Host Physical Address	CORE_ TA_UART1 Physical Address	Section
L4_TA_AGENT_CONTROL_H	RW	32	0x4806 3024	0x4806 5024	0x4806 B024	Section 9.5.2.4.2
L4_TA_AGENT_STATUS_L	RW	32	0x4806 3028	0x4806 5028	0x4806 B028	Section 9.5.2.4.3

Table 9-53. CORE_TA Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	CORE_TA_ USB_TLL Physical Address	CORE_TA_ USB_HS_HOST Physical Address	Section
L4_TA_AGENT_CONTROL_L	RW	32	0x4806 3020	0x4806 5020	Section 9.5.2.4.1
L4_TA_AGENT_CONTROL_H	RW	32	0x4806 3024	0x4806 5024	Section 9.5.2.4.2
L4_TA_AGENT_STATUS_L	RW	32	0x4806 3028	0x4806 5028	Section 9.5.2.4.3

Table 9-54. CORE_TA Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	CORE_ TA_UART2 Physical Address	CORE_ TA_I2C1 Physical Address	CORE_ TA_I2C12 Physical Address	Section
L4_TA_AGENT_CONTROL_L	RW	32	0x4806 D020	0x4807 1020	0x4807 1020	Section 9.5.2.4.1
L4_TA_AGENT_CONTROL_H	RW	32	0x4806 D024	0x4807 1024	0x4807 1024	Section 9.5.2.4.2
L4_TA_AGENT_STATUS_L	RW	32	0x4806 D028	0x4807 1028	0x4807 1028	Section 9.5.2.4.3

Table 9-55. CORE_TA Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	CORE_ TA_MCBSP1 Physical Address	CORE_TA_ GPTIMER10 Physical Address	CORE_TA_ GPTIMER11 Physical Address	Section
L4_TA_AGENT_CONTROL_L	RW	32	0x4807 5020	0x4808 7020	0x4809 9020	Section 9.5.2.4.1
L4_TA_AGENT_CONTROL_H	RW	32	0x4807 5024	0x4808 7024	0x4809 9024	Section 9.5.2.4.2
L4_TA_AGENT_STATUS_L	RW	32	0x4807 5028	0x4808 7028	0x4809 9028	Section 9.5.2.4.3

Table 9-56. CORE_TA Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	CORE_ TA_MAILBOX Physical Address	CORE_ TA_MCBSP5 Physical Address	Section
L4_TA_AGENT_CONTROL_L	RW	32	0x4809 5020	0x4809 9020	Section 9.5.2.4.1
L4_TA_AGENT_CONTROL_H	RW	32	0x4809 5024	0x4809 9024	Section 9.5.2.4.2
L4_TA_AGENT_STATUS_L	RW	32	0x4809 5028	0x4809 9028	Section 9.5.2.4.3

Table 9-57. CORE_TA Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	CORE_ TA_MCSPI1 Physical Address	CORE_ TA_MCSPI2 Physical Address	CORE_ TA_MMC1 Physical Address	Section
L4_TA_AGENT_CONTROL_L	RW	32	0x4809 9020	0x4809 B020	0x4809 D020	Section 9.5.2.4.1
L4_TA_AGENT_CONTROL_H	RW	32	0x4809 9024	0x4809 B024	0x4809 D024	Section 9.5.2.4.2

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 9-57. CORE_TA Common Register Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	CORE_ TA_MCSPI1 Physical Address	CORE_ TA_MCSPI2 Physical Address	CORE_ TA_MMC1 Physical Address	Section
L4_TA_AGENT_STATUS_L	RW	32	0x4809 9028	0x4809 B028	0x4809 D028	Section 9.5.2.4.3

Table 9-58. CORE_TA Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	CORE_ TA_MSPRO Physical Address	CORE_ TA_RNG Physical Address	Section
L4_TA_AGENT_CONTROL_L	RW	32	0x4809 F020	0x480A 1020	Section 9.5.2.4.1
L4_TA_AGENT_CONTROL_H	RW	32	0x4809 F024	0x480A 1024	Section 9.5.2.4.2
L4_TA_AGENT_STATUS_L	RW	32	0x4809 F028	0x480A 1028	Section 9.5.2.4.3

Table 9-59. CORE_TA Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	CORE_ TA_D3D1 Physical Address	CORE_ TA_AES1 Physical Address	Section
L4_TA_AGENT_CONTROL_L	RW	32	0x480A 3020	0x480A 7020	Section 9.5.2.4.1
L4_TA_AGENT_CONTROL_H	RW	32	0x480A 3024	0x480A 7024	Section 9.5.2.4.2
L4_TA_AGENT_STATUS_L	RW	32	0x480A 3028	0x480A 7028	Section 9.5.2.4.3

Table 9-60. CORE_TA Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	CORE_ TA_FPKA Physical Address	CORE_TA_ USB_HS_OTG Physical Address	Section
L4_TA_AGENT_CONTROL_L	RW	32	0x480A A020	0x480A C020	Section 9.5.2.4.1
L4_TA_AGENT_CONTROL_H	RW	32	0x480A A024	0x480A C024	Section 9.5.2.4.2
L4_TA_AGENT_STATUS_L	RW	32	0x480A A028	0x480A C028	Section 9.5.2.4.3

Table 9-61. CORE_TA Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	CORE_ TA_MMC3 Physical Address	TA_MG Physical Address	CORE_ TA_HDQ1 Physical Address	Section
L4_TA_AGENT_CONTROL_L	RW	32	0x480A E020	0x480B 1020	0x480B 3020	Section 9.5.2.4.1
L4_TA_AGENT_CONTROL_H	RW	32	0x480A E024	0x480B 1024	0x480B 3024	Section 9.5.2.4.2
L4_TA_AGENT_STATUS_L	RW	32	0x480A E028	0x480B 1028	0x480B 3028	Section 9.5.2.4.3

Table 9-62. CORE_TA Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	CORE_ TA_MMC2 Physical Address	CORE_ TA_ICR Physical Address	CORE_ TA_MCSPI3 Physical Address	Section
L4_TA_AGENT_CONTROL_L	RW	32	0x480B 5020	0x480B 7020	0x480B 9020	Section 9.5.2.4.1
L4_TA_AGENT_CONTROL_H	RW	32	0x480B 5024	0x480B 7024	0x480B 9024	Section 9.5.2.4.2
L4_TA_AGENT_STATUS_L	RW	32	0x480B 5028	0x480B 7028	0x480B 9028	Section 9.5.2.4.3

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 9-63. CORE_TA Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	CORE_TA_ MCSPI4 Physical Address	CORE_TA_ CAMERA Physical Address	CORE_TA_ D3D2 Physical Address	Section
L4_TA_AGENT_CONTROL_L	RW	32	0x480B B020	0x480C 0020	0x480C 2020	Section 9.5.2.4.1
L4_TA_AGENT_CONTROL_H	RW	32	0x480B B024	0x480C 0024	0x480C 2024	Section 9.5.2.4.2
L4_TA_AGENT_STATUS_L	RW	32	0x480B B028	0x480C 0028	0x480C 2028	Section 9.5.2.4.3

Table 9-64. CORE_TA Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	CORE_TA_ SHAM1 Physical Address	CORE_TA_ AES2 Physical Address	CORE_TA_ INTH Physical Address	Section
L4_TA_AGENT_CONTROL_L	RW	32	0x480C 4020	0x480C 6020	0x480C 8020	Section 9.5.2.4.1
L4_TA_AGENT_CONTROL_H	RW	32	0x480C 4024	0x480C 6024	0x480C 8024	Section 9.5.2.4.2
L4_TA_AGENT_STATUS_L	RW	32	0x480C 4028	0x480C 6028	0x480C 8028	Section 9.5.2.4.3

Table 9-65. CORE_TA Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	CORE_TA_ICR_ MODEM PhysicalAddress	Section
L4_TA_AGENT_CONTROL_L	RW	32	0x480C E020	Section 9.5.2.4.1
L4_TA_AGENT_CONTROL_H	RW	32	0x480C E024	Section 9.5.2.4.2
L4_TA_AGENT_STATUS_L	RW	32	0x480C E024	Section 9.5.2.4.3

Table 9-66. CORE_TA Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	CORE_TA_WKUP Physical Address	Section
L4_TA_AGENT_CONTROL_L	RW	32	0x4834 0020	Section 9.5.2.4.1
L4_TA_AGENT_CONTROL_H	RW	32	0x4834 0024	Section 9.5.2.4.2
L4_TA_AGENT_STATUS_L	RW	32	0x4834 0028	Section 9.5.2.4.3

Table 9-67. PER_TA Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	PER_TA_ UART3 Physical Address	PER_TA_ MCBSP2 Physical Address	PER_TA_ MCBSP3 Physical Address	Section
L4_TA_AGENT_CONTROL_L	RW	32	0x4902 1020	0x4902 3020	0x4902 5020	Section 9.5.2.4.1
L4_TA_AGENT_CONTROL_H	RW	32	0x4902 1024	0x4902 3024	0x4902 5024	Section 9.5.2.4.2
L4_TA_AGENT_STATUS_L	RW	32	0x4902 1028	0x4902 3028	0x4902 5028	Section 9.5.2.4.3

Table 9-68. PER_TA Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	PER_TA_ MCBSP4 Physical Address	PER_TA_ MCBSP2_ SIDETONE2 Physical Address	PER_TA_ MCBSP2_ SIDETONE3 Physical Address	Section
L4_TA_AGENT_CONTROL_L	RW	32	0x4902 7020	0x4902 9020	0x4902 B020	Section 9.5.2.4.1
L4_TA_AGENT_CONTROL_H	RW	32	0x4902 7024	0x4902 9024	0x4902 B024	Section 9.5.2.4.2

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 9-68. PER_TA Common Register Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	PER_TA_ MCBSP4 Physical Address	PER_TA_ MCBSP2_ SIDETONE2 Physical Address	PER_TA_ MCBSP2_ SIDETONE3 Physical Address	Section
L4_TA_AGENT_STATUS_L	RW	32	0x4902 7028	0x4902 9028	0x4902 B028	Section 9.5.2.4.3

Table 9-69. PER_TA Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	PER_TA_ WDTIMER3 Physical Address	PER_TA_ GPTIMER2 Physical Address	Section
L4_TA_AGENT_CONTROL_L	RW	32	0x4903 1020	0x4903 3020	Section 9.5.2.4.1
L4_TA_AGENT_CONTROL_H	RW	32	0x4903 1024	0x4903 3024	Section 9.5.2.4.2
L4_TA_AGENT_STATUS_L	RW	32	0x4903 1028	0x4903 3028	Section 9.5.2.4.3

Table 9-70. PER_TA Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	PER_TA_ GPTIMER3 Physical Address	PER_TA_ GPTIMER4 Physical Address	PER_TA_ GPTIMER5 Physical Address	Section
L4_TA_AGENT_CONTROL_L	RW	32	0x4903 5020	0x4903 7020	0x4903 9020	Section 9.5.2.4.1
L4_TA_AGENT_CONTROL_H	RW	32	0x4903 5024	0x4903 7024	0x4903 9024	Section 9.5.2.4.2
L4_TA_AGENT_STATUS_L	RW	32	0x4903 5028	0x4903 7028	0x4903 9028	Section 9.5.2.4.3

Table 9-71. PER_TA Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	PER_TA_ GPTIMER6 Physical Address	PER_TA_ GPTIMER7 Physical Address	PER_TA_ GPTIMER8 Physical Address	Section
L4_TA_AGENT_CONTROL_L	RW	32	0x4903 B020	0x4903 D020	0x4903 F020	Section 9.5.2.4.1
L4_TA_AGENT_CONTROL_H	RW	32	0x4903 B024	0x4903 D024	0x4903 F024	Section 9.5.2.4.2
L4_TA_AGENT_STATUS_L	RW	32	0x4903 B028	0x4903 D028	0x4903 F028	Section 9.5.2.4.3

Table 9-72. PER_TA Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	PER_TA_ GPTIMER9 Physical Address	PER_TA_ GPIO2 Physical Address	PER_TA_ GPIO3 Physical Address	Section
L4_TA_AGENT_CONTROL_L	RW	32	0x4904 1020	0x4905 1020	0x4905 3020	Section 9.5.2.4.1
L4_TA_AGENT_CONTROL_H	RW	32	0x4904 1024	0x4905 1024	0x4905 3024	Section 9.5.2.4.2
L4_TA_AGENT_STATUS_L	RW	32	0x4904 1028	0x4905 1028	0x4905 3028	Section 9.5.2.4.3

Table 9-73. PER_TA Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	PER_TA_ GPIO4 Physical Address	PER_TA_ GPIO5 Physical Address	PER_TA_ GPIO6 Physical Address	Section
L4_TA_AGENT_CONTROL_L	RW	32	0x4905 5020	0x4905 7020	0x4905 9020	Section 9.5.2.4.1
L4_TA_AGENT_CONTROL_H	RW	32	0x4905 5024	0x4905 7024	0x4905 9024	Section 9.5.2.4.2

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 9-73. PER_TA Common Register Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	PER_TA_ GPIO4 Physical Address	PER_TA_ GPIO5 Physical Address	PER_TA_ GPIO6 Physical Address	Section
L4_TA_AGENT_STATUS_L	RW	32	0x4905 5028	0x4905 7028	0x4905 9028	Section 9.5.2.4.3

Table 9-74. EMU_TA Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	EMU_TA_ TESTCHIPLEVELTAP Physical Address	EMU_TA_ MPU Physical Address	EMU_TA_ TPUI Physical Address	Section
L4_TA_AGENT_CONTROL_L	RW	32	0x5400 5020	0x5401 8020	0x5401 A020	Section 9.5.2.4.1
L4_TA_AGENT_CONTROL_H	RW	32	0x5400 5024	0x5401 8024	0x5401 A024	Section 9.5.2.4.2
L4_TA_AGENT_STATUS_L	RW	32	0x5400 5028	0x5401 8028	0x5401 A028	Section 9.5.2.4.3

Table 9-75. EMU_TA Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	EMU_TA_ ETB Physical Address	EMU_TA_ DAPCTL Physical Address	EMU_TA_ SDTI Physical Address	EMU_TA_ L4WKUP Physical Address	Section
L4_TA_AGENT_CONTROL_L	RW	32	0x5401 C020	0x5401 E020	0x5401 F020	0x5473 0020	Section 9.5.2.4.1
L4_TA_AGENT_CONTROL_H	RW	32	0x5401 C024	0x5401 E024	0x5401 F024	0x5473 0024	Section 9.5.2.4.2
L4_TA_AGENT_STATUS_L	RW	32	0x5401 C028	0x5401 E028	0x5401 F028	0x5473 0028	Section 9.5.2.4.3

Table 9-76. WKUP_TA Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	WKUP_TA_ GPTIMER12 Physical Address	WKUP_TA_ PRM Physical Address	WKUP_TA_ WDTIMER1 Physical Address	Section
L4_TA_AGENT_CONTROL_L	RW	32	0x4830 5020	0x4830 9020	0x4830 D020	Section 9.5.2.4.1
L4_TA_AGENT_CONTROL_H	RW	32	0x4830 5024	0x4830 9024	0x4830 D024	Section 9.5.2.4.2
L4_TA_AGENT_STATUS_L	RW	32	0x4830 5028	0x4830 9028	0x4830 D028	Section 9.5.2.4.3

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 9-77. WKUP_TA Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	WKUP_TA_ USIM Physical Address	WKUP_TA_ GPIO1 Physical Address	WKUP_TA_ WDTIMER Physical Address	Section
L4_TA_AGENT_CONTROL_L	RW	32	0x4830 F020	0x4831 1020	0x4831 5020	Section 9.5.2.4.1
L4_TA_AGENT_CONTROL_H	RW	32	0x4830 F024	0x4831 1024	0x4831 5024	Section 9.5.2.4.2
L4_TA_AGENT_STATUS_L	RW	32	0x4830 F028	0x483 1028	0x4831 5028	Section 9.5.2.4.3

Table 9-78. WKUP_TA Common Register Mapping Summary

Register Name	Type	Register Width (Bits)	WKUP_TA_ GPTIMER1 Physical Address	WKUP_TA_ SYNCTIMER32K Physical Address	Section
L4_TA_AGENT_CONTROL_L	RW	32	0x4831 9020	0x4832 1020	Section 9.5.2.4.1
L4_TA_AGENT_CONTROL_H	RW	32	0x4831 9024	0x4832 1024	Section 9.5.2.4.2
L4_TA_AGENT_STATUS_L	RW	32	0x4831 9028	0x4832 1028	Section 9.5.2.4.3

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.2.4 L4 Target Agent (L4 TA) Register Descriptions

9.5.2.4.1 L4_TA_AGENT_CONTROL_L

This register enables error reporting. The L4_TA_AGENT_CONTROL_L is shown in Figure 9-26 and described in Table 9-79.

Figure 9-26. L4_TA_AGENT_CONTROL_L

31 – 25	24
Reserved	SERROR_REP
	R

23 – 11	10 – 8	7 – 1	0
Reserved	REQ_TIMEOUT	Reserved	OCP_RESET
	RW		R/W

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-79. L4_TA_AGENT_CONTROL_L Field Descriptions

Bit	Field	Value	Description
31-25	Reserved	0	Read returns 0.
24	SERROR_REP		Enable logging of error
23-11	Reserved	0	Read returns 0.
10-8	REQ_TIMEOUT		Timeout Bound. Values are:0 - No timeout
		1	1x base cycles
		2	4x base cycles
		3	16x base cycles
		4	64x base cycles
7-1	Reserved		Read returns 0.
0	OCP_RESET		The OCP_RESET field controls the OCP reset signal to the attached core. Setting this bit clears any pending transfers and resets the OCP interface. The bit must be cleared to de-assert the OCP reset signal. When the software reset feature is available on a target agent, the target agent OCP must also have a reset signal directed to the target core.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.2.4.2 L4_TA_AGENT_CONTROL_H

This register enables clock power management. The L4_TA_AGENT_CONTROL_H is shown in Figure 9-27 and described in Table 9-80.

Figure 9-27. L4_TA_AGENT_CONTROL_H

31			16
Reserved			

15	9	8	7	0
Reserved		EXT_CLOCK	Reserved	
		R		

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-80. L4_TA_AGENT_CONTROL_H Field Descriptions

Bit	Field	Value	Description
31-9	Reserved	0	Read returns 0.
8	EXT_CLOCK		When set to 1, the ext_clk_off_i signal on a target agent indicates when the target agent should shut off.
7-0	Reserved	0	Read returns 0.

9.5.2.4.3 L4_TA_AGENT_STATUS_L

This register enables error reporting. The L4_TA_AGENT_STATUS_L is shown in Figure 9-28 and described in Table 9-81.

Figure 9-28. L4_TA_AGENT_STATUS_L

31			16
Reserved			

15	9	8	7	0
Reserved		REQ_TIMEOUT	Reserved	
		R 1toCLR		

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-81. L4_TA_AGENT_STATUS_L Field Descriptions

Bit	Field	Value	Description
31-9	Reserved	0	Read returns 0.
8	REQ_TIMEOUT	0x0 0x1	No request timeout A request timeout has occurred
7-0	Reserved	0	Read returns 0.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.2.5 L4 Link Register Agent (LA) Register Mapping Summary

This section provides information on the L4-Core link agent (LA) register module. Each of the registers within the module instance is described separately in Table 9-82.

The LA register block contains the initiator subsystem information register and the composite sideband signal mask and status registers.

Table 9-82. L4 LA Register Mapping Summary

Register Name	Type	Register Width (Bits)	CORE_LA Physical Address	PER_LA Physical Address	EMU_LA Physical Address	WKUP_LA Physical Address	
L4_LA_NETWORK_H	R	32	0x4804 1014	0x4900 1014	0x5400 7014	0x4832 9014	Section 9.5.2.6.1
L4_LA_INITIATOR_INFO_L	R	32	0x4804 1018	0x4900 1018	0x5400 7018	0x4832 9018	Section 9.5.2.6.2
L4_LA_INITIATOR_INFO_H	R	32	0x4804 101C	0x4900 101C	0x5400 701C	0x4832 901C	Section 9.5.2.6.3
L4_LA_NETWORK_CONTROL_L	RW	32	0x4804 1020	0x4900 1020	0x5400 7020	0x4832 9020	Section 9.5.2.6.4
L4_LA_NETWORK_CONTROL_H	RW	32	0x4804 1024	0x4900 1024	0x5400 7024	0x4832 9024	Section 9.5.2.6.5

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.2.6 L4 Link Register Agent (LA) Register Descriptions

9.5.2.6.1 L4_LA_NETWORK_H

This register identifies the interconnect. The L4_LA_NETWORK_H is shown in Figure 9-29 and described in Table 9-83.

Figure 9-29. L4_LA_NETWORK_H

31	0
ID	
R	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-83. L4_LA_NETWORK_H Field Descriptions

Bit	Field	Value	Description
31-0	ID		The ID field uniquely identifies this interconnect, and can serve as a chip ID.

9.5.2.6.2 L4_LA_INITIATOR_INFO_L

This register contains initiator subsystem information. The L4_LA_INITIATOR_INFO_L is shown in Figure 9-30 and described in Table 9-84.

Figure 9-30. L4_LA_INITIATOR_INFO_L

31–28	27–24	23–16
Reserved	PROT_GROUPS	NUMBER_REGIONS
	R	R

15–4	3–0
Reserved	SEGMENTS
	R

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-84. L4_LA_INITIATOR_INFO_L Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Read returns 0.
27-24	PROT_GROUPS		The number of protection groups. The PROT_GROUPS field contains read-only configuration information for the address mapping and security structure of the initiator subsystem. If the PROT_GROUPS field is set to 0, there are no protection group registers.
23-16	NUMBER_REGIONS		The number of regions. The NUMBER_REGIONS field contains read-only configuration information for the region register of the initiator subsystem.
15-4	Reserved	0	Read returns 0.
3-0	SEGMENTS		The number of segments. The SEGMENT fields contains read-only configuration information for the segment register of the initiator subsystem.

Table 9-85. Reset value for L4_LA_INITIATOR_INFO_L

Field Name	CORE_LA	PER_LA	EMU_LA	WKUP_LA
PROT_GROUPS	0x8	0x8	0x8	0x8
NUMBER_REGIONS	0x64	0x2B	0x1A	0x13
SEGMENTS	0x6	0x5	0x3	0x2

9.5.2.6.3 L4_LA_INITIATOR_INFO_H

This register contains initiator subsystem information. The L4_LA_INITIATOR_INFO_H is shown in Figure 9-31 and described in Table 9-86.

Copyright © 2011, Texas Instruments Incorporated
DRAFT ONLY

Figure 9-31. L4_LA_INITIATOR_INFO_H

31–19	18–16
Reserved	THREADS
	R

15	14–12	11	10–8	7–5	4–0
Reserved	CONNID_WIDTH	Reserved	BYTE_DATA_WIDTH_EXP	Reserved	ADDR_WIDTH
	R		R		R

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-86. L4_LA_INITIATOR_INFO_H Field Descriptions

Bit	Field	Value	Description
31-19	Reserved	0	Read returns 0.
18-16	THREADS		The THREADS field specifies the number of initiator threads connected to the interconnect. The field contains read-only configuration information for the initiator subsystem.
15	Reserved	0	Read returns 0.
14-12	CONNID_WIDTH		The initiator subsystem connID width. The CONNID_WIDTH field contains read-only configuration information for the initiator subsystem.
11	Reserved	0	Read returns 0.
10-8	BYTE_DATA_WIDTH_EXP		This field specifies the initiator subsystem data width. 1:2^1 bytes specifies a 16-bit data width and 2:2^2 bytes specifies a 32-bit data width. The BYTE_DATA_WIDTH_EXP field contains read-only configuration information for the initiator subsystem.
7-5	Reserved	0	Read returns 0.
4-0	ADDR_WIDTH		This field specifies the initiator subsystem address width. The ADDR_WIDTH field contains read-only configuration information for the initiator subsystem.

Table 9-87. Reset value for L4_LA_INITIATOR_INFO_H

Field Name	CORE_LA	PER_LA	EMU_LA	WKUP_LA
THREADS	0x4	0x4	0x2	0x2
CONNID_WIDTH	0x4	0x4	0x4	0x0
BYTE_DATA_WIDTH_EXP	0x2	0x2	0x2	0x2
ADDR_WIDTH	0x18	0x14	0x18	0x14

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.2.6.4 L4_LA_NETWORK_CONTROL_L

This register controls interconnect minimum timeout values. The L4_LA_NETWORK_CONTROL_L is shown in Figure 9-32 and described in Table 9-88.

Figure 9-32. L4_LA_NETWORK_CONTROL_L

31		16
Reserved		

15	11	10	8	7	0
Reserved		TIMEOUT_BASE		Reserved	
		R/W			

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-88. L4_LA_NETWORK_CONTROL_L Field Descriptions

Bit	Field	Value	Description
31-11	Reserved	0	Read returns 0.
10-8	TIMEOUT_BASE		The TIMEOUT_BASE field indicates the timeout period (that is, base cycles) for the highest frequency time-base signal sent from the L4 initiator subsystem to all target agents that have timeout enabled. Values for the field are:
		0	Timeout disabled
		1	L4 interconnect clock cycles divided by 64
		2	L4 interconnect clock cycles divided by 256
		3	L4 interconnect clock cycles divided by 1024
		4	L4 interconnect clock cycles divided by 4096
7-0	Reserved	0	Read returns 0.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.2.6.5 L4_LA_NETWORK_CONTROL_H

This register controls interconnect global power control. The L4_LA_NETWORK_CONTROL_H is shown in Figure 9-33 and described in Table 9-89.

Figure 9-33. L4_LA_NETWORK_CONTROL_H

31–25	24	23–21	20	19–16
Reserved	CLOCK_GATE_DISABLE	Reserved	THREAD0_PRI	Reserved
	R/W		R	

15–9	8	7–0
Reserved	EXT_CLOCK	Reserved
	R	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-89. L4_LA_NETWORK_CONTROL_H Field Descriptions

Bit	Field	Value	Description
31-25	Reserved	0	Read returns 0.
24	CLOCK_GATE_DISABLE		When set to 1 this field disables all clock gating.
23-21	Reserved	0	Read returns 0.
20	THREAD0_PRI		Sets thread priority. If the field is set to 0, the default, all initiator threads are treated the same. Setting the THREAD0_PRI field to 1 assigns a higher arbitration priority to thread 0 of the first initiator OCP interface. To avoid starvation, arbitration is imposed by the initiator subsystem. When multiple requests from different initiator threads are dispatched to targets simultaneously, the oldest request is dispatched first. If thread 0 is assigned a higher priority, a request on thread 0 always wins arbitration. Assigning thread 0 of the first initiator OCP the highest priority on a request or response can result in the starvation of other threads.
19-9	Reserved	0	Read returns 0.
8	EXT_CLOCK		When set to 1, the ext_clk_off_i signal on the initiator subsystem instructs the entire L4 to shut off.
7-0	Reserved	0	Read returns 0.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.2.7 L4 Address Protection (AP) Register Mapping Summary

This section provides information on the L4-Core link agent (LA) register module. Each of the registers within the module instance is described separately in Table 9-90.

The AP register block contains the segment, address region, and protection group registers that can be used to specify the addressing scheme or restrict the access of target address regions.

Table 9-90. L4 AP Register Mapping Summary

Register Name	Type	Register Width (Bits)	CORE_AP Physical Address	PER_AP Physical Address	Section
L4_AP_SEGMENT_i_L[(1)]	RW	32	0x4804 0100 + (0x08*i)	0x4900 0100 + (0x08*i)	Section 9.5.2.8.1
L4_AP_SEGMENT_i_H[(1)]	RW	32	0x4804 0104 + (0x08*i)	0x4900 0104 + (0x08*i)	Section 9.5.2.8.2
L4_AP_PROT_GROUP_MEMBERS_k_L[(2)]	R	32	0x4804 0200 + (0x08*k)	0x4900 0200 + (0x08*k)	Section 9.5.2.8.3
L4_AP_PROT_GROUP_ROLES_k_L[(2)]	R	32	0x4804 0280 + (0x08*k)	0x4900 0280 + (0x08*k)	Section 9.5.2.8.4
L4_AP_REGION_l_L[(3)]	RW	32	0x4804 0300 + (0x08*l)	0x4900 0300 + (0x08*l)	Section 9.5.2.8.5
L4_AP_REGION_l_H[(3)]	RW	32	0x4804 0304 + (0x08*l)	0x4900 0304 + (0x08*l)	Section 9.5.2.8.6

(1) i = 0 to 5 for CORE_AP,
i = 0 to 4 for PER_AP,
i = 0 to 2 for EMU_AP,
i = 0 to 1 for WKUP_AP,

(2) k = 0 to 7 for CORE_ AP and PER_AP.
k = 0 to 5 for EMU_AP

(3) l = 0 to 99 for CORE_AP,
l = 0 to 42 for PER_AP,
l = 0 to 25 for EMU_AP,
l = 0 to 18 for WKUP_AP,

Table 9-91. L4 AP Register Mapping Summary

Register Name	Type	Register Width (Bits)	EMU_AP Physical Address	WKUP_AP Physical Address	Section
L4_AP_SEGMENT_i_L[(1)]	RW	32	0x5400 6100 + (0x08*i)	0x4832 8100 + (0x08*i)	Section 9.5.2.8.1
L4_AP_SEGMENT_i_H[(1)]	RW	32	0x5400 6104 + (0x08*i)	0x4832 8104 + (0x08*i)	Section 9.5.2.8.2
L4_AP_PROT_GROUP_MEMBERS_k_L[(2)]	R	32	0x5400 6200 + (0x08*k)	N/A	Section 9.5.2.8.3
L4_AP_PROT_GROUP_ROLES_k_L[(2)]	R	32	0x5400 6280 + (0x08*k)	N/A	Section 9.5.2.8.4
L4_AP_REGION_l_L[(3)]	RW	32	0x5400 6300 + (0x08*l)	0x4832 8300 + (0x08*l)	Section 9.5.2.8.5
L4_AP_REGION_l_H[(3)]	RW	32	0x5400 6304 + (0x08*l)	0x4832 8304 + (0x08*l)	Section 9.5.2.8.6

(1) i = 0 to 5 for CORE_AP,
i = 0 to 4 for PER_AP,
i = 0 to 2 for EMU_AP,
i = 0 to 1 for WKUP_AP,

(2) k = 0 to 7 for CORE_ AP and PER_AP.
k = 0 to 5 for EMU_AP

(3) l = 0 to 99 for CORE_AP,
l = 0 to 42 for PER_AP,
l = 0 to 25 for EMU_AP,
l = 0 to 18 for WKUP_AP,

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.2.7.1 Reset Values

Table 9-92. Reset Values for CORE_AP L4_AP_REGION_I_L and L4_AP_REGION_I_H

y	BASE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	SIZE
0	0x00 0000	1	0	2	0x0B
1	0x00 0800	1	7	2	0x0B
2	0x00 1000	1	7	2	0x0C
3	0x01 0000	1	2	2	0x0A
4	0x01 0400	1	2	2	0x0A
5	0x01 0800	1	2	2	0x0A
6	0x01 0C00	1	2	2	0x0A
7	0x01 1000	1	2	2	0X0C
8	0x00 0000	3	3	2	0x0C
9	0x01 6000	1	7	2	0x0C
10	0x01 7000	1	7	2	0x0C
11	0x01 8000	1	7	2	0x0C
12	0x01 C000	1	7	2	0x0C
13	0x02 B000	2	7	2	0x0C
14	0x02 C000	2	7	2	0x0C
15	0x01 E000	1	7	2	0x0C
16	0x01 F000	1	7	2	0x0C
17	0x02 A000	1	7	2	0x0C
18	0x02 B000	1	7	2	0x0C
19	0x02 C000	1	7	2	0X0C
20	0x02 D000	1	7	2	0x0C
21	0x03 0000	1	7	1	0x0C
22	0x03 1000	1	7	1	0x0C
23	0x03 2000	1	7	1	0x0C
24	0x03 3000	1	7	1	0x0C
25	0x03 4000	1	7	2	0x0C
26	0x03 5000	1	7	2	0x0C
27	0x00 6000	2	7	2	0x0C
28	0x00 7000	2	7	2	0x0C
29	0x00 8000	2	7	2	0x0C
30	0x00 9000	2	7	2	0x0C
31	0x00 9000	3	7	2	0x0C
32	0x00 A000	3	7	2	0x0C
33	0x01 2000	2	7	2	0x0C
34	0x01 3000	2	7	2	0x0C
35	0x01 4000	2	7	2	0x0C
36	0x01 5000	2	7	2	0x0C
37	0x01 8000	2	7	2	0x0C
38	0x01 9000	2	7	2	0x0C
39	0x01 A000	2	7	2	0x0C
40	0x01 B000	2	7	2	0x0C
41	0x01 C000	2	7	2	0x0C
42	0x01 D000	2	7	2	0x0C
43	0x03 4000	2	7	2	0x0C
44	0x03 5000	2	7	2	0x0C
45	0x01 E000	2	7	2	0x0C

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 9-92. Reset Values for CORE_AP L4_AP_REGION_I_L and L4_AP_REGION_I_H (continued)

y	BASE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	SIZE
46	0x01 F000	2	7	2	0x0C
47	0x02 0000	2	1	2	0x0C
48	0x02 1000	2	1	2	0x0C
49	0x02 2000	2	1	2	0x0C
50	0x02 3000	2	1	2	0x0C
51	0x02 4000	2	1	2	0x0C
52	0x02 5000	2	1	2	0x0C
53	0x02 6000	2	1	2	0x0C
54	0x02 7000	2	1	2	0x0C
55	0x02 8000	2	1	2	0x0D
56	0x02 A000	2	1	2	0x0C
57	0x03 0000	2	7	2	0x0C
58	0x03 1000	2	7	2	0x0C
59	0x03 2000	2	7	2	0x0C
60	0x03 3000	2	7	2	0x0C
61	0x01 6000	2	7	2	0x0C
62	0x01 7000	2	7	2	0x0C
63	0x01 9000	1	7	2	0x0C
64	0x01 A000	1	7	2	0x0C
65	0x01 B000	1	7	2	0x0C
66	0x03 8000	2	7	2	0x0C
67	0x03 9000	2	7	2	0x0C
68	0x00 4000	0	7	2	0x0C
69	0x00 7000	0	7	2	0x0C
70	0x00 B000	3	7	2	0x0C
71	0x00 C000	3	7	2	0x0C
72	0x00 6000	0	7	2	0x0B
73	0x02 0000	1	7	1	0x0C
74	0x02 1000	1	7	1	0x0C
75	0x00 2000	0	7	2	0x0C
76	0x00 3000	0	7	2	0x0C
77	0x03 C000	2	3	3	0x0C
78	0x00 4000	4	4	2	0x0D
79	0x03 A000	2	7	2	0x0C
80	0x03 B000	2	7	2	0x0C
81	0x00 0000	5	7	2	0x0C
82	0x00 1000	3	1	2	0x0C
83	0x00 2000	3	1	2	0x0C
84	0x00 3000	3	1	2	0x0C
85	0x00 4000	3	1	2	0x0C
86	0x00 5000	3	1	2	0x0C
87	0x00 6000	3	1	2	0x0C
88	0x03 6000	2	7	2	0x0C
89	0x03 7000	2	7	2	0x0C
90	0x00 7000	3	7	2	0x0C
91	0x00 8000	3	7	2	0x0C
92	0x00 D000	3	7	2	0x0C

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 9-92. Reset Values for CORE_AP L4_AP_REGION_I_L and L4_AP_REGION_I_H (continued)

y	BASE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	SIZE
93	0x00 E000	3	7	2	0x0C
94	0x00 6000	4	7	2	0x0D
95	0x00 8800	4	7	2	0x0B
96	0x00 9000	4	7	2	0x0C
97	0x00 C000	4	1	2	0x0D
98	0x01 0000	4	7	2	0x10
99	0x02 0000	4	7	2	0x10

Table 9-93. Reset Values for PER_AP L4_AP_REGION_I_L and L4_AP_REGION_I_H

y	BASE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	SIZE
0	0x00 0000	0	0	2	0x0B
1	0x00 0800	0	7	2	0x0B
2	0x00 1000	0	7	2	0x0C
3	0x00 0000	1	7	2	0x0C
4	0x00 1000	1	7	2	0X0C
5	0x00 2000	1	7	2	0X0C
6	0x00 3000	1	7	2	0X0C
7	0x00 4000	1	7	2	0X0C
8	0x00 5000	1	7	2	0x0C
9	0x00 6000	1	7	2	0x0C
10	0x00 7000	1	7	2	0x0C
11	0x00 0000	1	7	2	0x0C
12	0x00 1000	2	7	2	0x0C
13	0x00 2000	2	7	2	0x0C
14	0x00 3000	2	7	2	0x0C
15	0x00 4000	2	7	2	0x0C
16	0x00 5000	2	7	2	0x0C
17	0x00 6000	2	7	2	0x0C
18	0x00 7000	2	7	2	0x0C
19	0x00 8000	2	7	2	0X0C
20	0x00 9000	2	7	2	0x0C
21	0x00 A000	2	7	1	0x0C
22	0x00 B000	2	7	2	0x0C
23	0x00 C000	2	7	2	0x0C
24	0x00 D000	2	7	2	0x0C
25	0x00 E000	2	7	2	0x0C
26	0x00 F000	2	7	2	0x0C
27	0x00 0000	3	7	2	0x0C
28	0x00 1000	3	7	2	0x0C
29	0x00 0000	4	7	2	0x0C
30	0x00 1000	4	7	2	0x0C
31	0x00 2000	4	7	2	0x0C
32	0x00 3000	4	7	2	0x0C
33	0x00 4000	4	7	2	0x0C
34	0x00 5000	4	7	2	0x0C
35	0x00 6000	4	7	2	0x0C

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 9-93. Reset Values for PER_AP L4_AP_REGION_I_L and L4_AP_REGION_I_H (continued)

y	BASE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	SIZE
36	0x00 7000	4	7	2	0x0C
37	0x00 8000	4	7	2	0x0C
38	0x00 9000	4	7	2	0x0C
39	0x00 8000	4	7	2	0x0C
40	0x00 9000	1	7	2	0x0C
41	0x00 A000	1	7	2	0x0C
42	0x00 B000	1	7	2	0x0C

Table 9-94. Reset Values for EMU_AP L4_AP_REGION_I_L and L4_AP_REGION_I_H

y	BASE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	SIZE
0	0x00 0000	2	5	2	0x14
1	0x00 4000	0	5	2	0x0C
2	0x00 5000	0	5	2	0x0C
3	0x00 6000	0	5	0	0x0B
4	0x00 6800	0	5	2	0x0B
5	0x00 7000	0	5	2	0x0B
6	0x00 8000	0	5	2	0x0C
7	0x01 8000	0	3	2	0x0C
8	0x01 9000	0	3	2	0x0C
9	0x01 A000	0	3	2	0x0C
10	0x01 B000	0	3	2	0x0C
11	0x01 C000	0	3	2	0x0C
12	0x01 D000	0	3	2	0x0C
13	0x01 E000	0	3	2	0x0C
14	0x10 0000	1	5	2	0x0C
15	0x01 F000	0	5	2	0x0C
16	0x01 0000	0	3	2	0x0E
17	0x10 6000	2	5	2	0x0C
18	0x10 8000	2	5	2	0x0C
19	0x10 4000	2	2	2	0X0D
20	0x10 8800	2	5	2	0x0B
21	0x10 9000	2	5	2	0x0C
22	0x10C000	2	1	2	0x0D
23	0x11 0000	2	5	2	0x10
24	0x12 0000	2	5	2	0x10
25	0x13 0000	2	5	2	0x0C

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 9-95. Reset Values for WKPUP_AP L4_AP_REGION_I_L and L4_AP_REGION_I_H

y	BASE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	SIZE
0	0x00 0000	1	0	2	0x0B
1	0x00 6000	0	0	2	0x0D
2	0x00 9000	0	0	2	0x0C
3	0x00 C000	0	0	2	0x0C
4	0x00 D000	0	0	2	0x0C
5	0x01 8000	0	0	2	0x0C
6	0x01 9000	0	0	2	0x0C
7	0x01 4000	0	0	2	0x0C
8	0x01 5000	0	0	2	0x0C
9	0x00 9000	1	0	2	0x0C
10	0x00 4000	0	0	2	0x0C
11	0x00 5000	0	0	2	0x0C
12	0x00 8800	1	0	2	0x0B
13	0x00 8000	0	0	2	0x0B
14	0x01 0000	0	0	2	0x0C
15	0x01 1000	0	0	2	0x0C
16	0x00 0000	1	0	2	0x0C
17	0x00 1000	1	0	2	0x0C
18	0x00 A000	1	0	2	0x0C

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.2.8 L4 Address Protection (AP) Register Descriptions

9.5.2.8.1 L4_AP_SEGMENT_i_L

This register defines the base address of each segments. The L4_AP_SEGMENT_i_L is shown in Figure 9-34 and described in Table 9-96.

Figure 9-34. L4_AP_SEGMENT_i_L

31–24	23–16
Reserved	BASE
	R

15–0
BASE
R

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-96. L4_AP_SEGMENT_i_L Field Descriptions

Bit	Field	Value	Description
31-24	Reserved	0	Read returns 0.
23-0	BASE		The base address of the segment (with 0s from bit 0 to bit SIZE-1).

Table 9-97. Reset Values for L4_AP_SEGMENT_i_L

BASE	CORE_AP	PER_AP	EMU_AP	WKUP_AP
i = 0	0x00 0000	0x00 0000	0x00 0000	0x00 0000
i = 1	0x04 0000	0x02 0000	0x40 0000	0x02 0000
i = 2	0x08 0000	0x03 0000	0x60 0000	N/A
i = 3	0x0C 0000	0x04 0000	N/A	N/A
i = 4	0x30 0000	0x05 0000	N/A	N/A
i = 5	0x32 0000	N/A	N/A	N/A

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.2.8.2 L4_AP_SEGMENT_i_H

This register defines the size of each segments. The L4_AP_SEGMENT_i_H is shown in Figure 9-35 and described in Table 9-98.

Figure 9-35. L4_AP_SEGMENT_i_H

31	16
Reserved	

15	5	4	0
Reserved		SIZE	
		R	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-98. L4_AP_SEGMENT_i_H Field Descriptions

Bit	Field	Value	Description
31-5	Reserved	0	Read returns 0.
4-0	SIZE		Segment size is a power of 2, where 2^SIZE is the byte size of a segment (all segment registers use the same size).

Table 9-99. Reset Values for L4_AP_SEGMENT_i_H

SIZE	CORE_AP	PER_AP	EMU_AP	WKUP_AP
For all i	0x12	0x10	0x15	0x11

9.5.2.8.3 L4_AP_PROT_GROUP_MEMBERS_k_L

This register defines connID bit vectors for a protection group. The L4_AP_PROT_GROUP_MEMBERS_k_L is shown in Figure 9-36 and described in Table 9-100.

Figure 9-36. L4_AP_PROT_GROUP_MEMBERS_k_L

31	16
Reserved	

15	0
CONNID_BIT_VECTOR	
R	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-100. L4_AP_PROT_GROUP_MEMBERS_k_L Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Read returns 0.
15-0	CONNID_BIT_VECTOR		A bit of 1 in position n means that connID n is allowed in this protection group. Illegal connIDs have their bits set to 0s.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.2.8.4 L4_AP_PROT_GROUP_ROLES_k_L

This register defines MReqInfo bit vectors for a protection group. The L4_AP_PROT_GROUP_ROLES_k_L is shown in Figure 9-37 and described in Table 9-101.

Figure 9-37. L4_AP_PROT_GROUP_ROLES_k_L

31	16
Reserved	

15	0
ENABLE	
R	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-101. L4_AP_PROT_GROUP_ROLES_k_L Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Read returns 0.
15-0	ENABLE		Setting of type access allowed for the group of initiators.

9.5.2.8.5 L4_AP_REGION_l_L

This register defines the base address of the region in respect to the segment it belongs to. The L4_AP_REGION_l_L is shown in Figure 9-38 and described in Table 9-102.

Figure 9-38. L4_AP_REGION_l_L

31	24	23	16
Reserved		BASE	
		R	

15	0
BASE	
R	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-102. L4_AP_REGION_l_L Field Descriptions

Bit	Field	Value	Description
31-24	Reserved	0	Read returns 0.
23-0	BASE		Sets the base address of this region relative to its segment base.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

9.5.2.8.6 L4_AP_REGION_l_H

This register defines the size, protection group^and segment ID of the region. The L4_AP_REGION_l_H is shown in Figure 9-39 and described in Table 9-103.

Figure 9-39. L4_AP_REGION_l_H

31–28	27–24	23	22–20	19–17	16
Reserved	SEGMENT_ID	Reserved	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	Reserved
	R		R/W	R	

15–6	5–1	0
Reserved	SIZE	ENABLE
	R	R

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 9-103. L4_AP_REGION_l_H Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Read returns 0
27-24	SEGMENT_ID		Identifies the segment to which the region is part of.
23	Reserved	0	Read returns 0
22-20	PROT_GROUP_ID		The protection group containing this region.
19-17	BYTE_DATA_WIDTH_EXP		The target OCP data byte width is 2^(BYTE_DATA_WIDTH_EXP) bytes. The value of this field is derived from the target OCP data_width parameter.
16-6	Reserved	0	Read returns 0
5-1	SIZE		Define the size of the region in bytes. 2^SIZE equals the region.
0	ENABLE	0x0 0x1	Disable the region, no access allows Enable the region, with access as define in registers

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Chapter 10

SPRUH73–July 2011

Enhanced Direct Memory Access (EDMA)

This chapter describes the EDMA of the device.

Topic **Page**

10.1 Introduction **1360**
10.2 Integration **1360**
10.3 Functional Description **1360**
10.4 Use Cases **1360**
10.5 EDMA3 Registers **1361**

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.1 Introduction

10.2 Integration

10.3 Functional Description

10.4 Use Cases

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5 EDMA3 Registers

This section discusses the registers of the EDMA3 controller.

10.5.1 Register Memory Maps

See your device-specific data manual for the register memory maps.

10.5.2 EDMA3 Channel Controller Control Registers

Table 10-1 lists the memory-mapped registers for the EDMA3 channel controller (EDMACC). See the device-specific data manual for the memory address of these registers, and for the shadow region addresses. All other register offset addresses not listed in Table 10-1 should be considered as reserved locations and the register contents should not be modified.

Table 10-1. EDMACC Registers

Offset	Acronym	Register Description	Section
00h	PID	Peripheral Identification Register	Section 10.5.2.1.1
04h	CCCFG	EDMA3CC Configuration Register	Section 10.5.2.1.2
0100h-01FCh	DCHMAP0-63	DMA Channel 0-63 Mapping Registers	Section 10.5.2.1.3
0200h	QCHMAP0	QDMA Channel 0 Mapping Register	Section 10.5.2.1.4
0204h	QCHMAP1	QDMA Channel 1 Mapping Register	Section 10.5.2.1.4
0208h	QCHMAP2	QDMA Channel 2 Mapping Register	Section 10.5.2.1.4
020Ch	QCHMAP3	QDMA Channel 3 Mapping Register	Section 10.5.2.1.4
0210h	QCHMAP4	QDMA Channel 4 Mapping Register	Section 10.5.2.1.4
0214h	QCHMAP5	QDMA Channel 5 Mapping Register	Section 10.5.2.1.4
0218h	QCHMAP6	QDMA Channel 6 Mapping Register	Section 10.5.2.1.4
021Ch	QCHMAP7	QDMA Channel 7 Mapping Register	Section 10.5.2.1.4
0240h	DMAQNUM0	DMA Queue Number Register 0	Section 10.5.2.1.5
0244h	DMAQNUM1	DMA Queue Number Register 1	Section 10.5.2.1.5
0248h	DMAQNUM2	DMA Queue Number Register 2	Section 10.5.2.1.5
024Ch	DMAQNUM3	DMA Queue Number Register 3	Section 10.5.2.1.5
0250h	DMAQNUM4	DMA Queue Number Register 4	Section 10.5.2.1.5
0254h	DMAQNUM5	DMA Queue Number Register 5	Section 10.5.2.1.5
0258h	DMAQNUM6	DMA Queue Number Register 6	Section 10.5.2.1.5
025Ch	DMAQNUM7	DMA Queue Number Register 7	Section 10.5.2.1.5
0260h	QDMAQNUM	QDMA Queue Number Register	Section 10.5.2.1.6
0284h	QUEPRI	Queue Priority Register	Section 10.5.2.1.7
0300h	EMR	Event Missed Register	Section 10.5.2.2.1
0304h	EMRH	Event Missed Register High	Section 10.5.2.2.1
0308h	EMCR	Event Missed Clear Register	Section 10.5.2.2.2
030Ch	EMCRH	Event Missed Clear Register High	Section 10.5.2.2.2
0310h	QEMR	QDMA Event Missed Register	Section 10.5.2.2.3
0314h	QEMCR	QDMA Event Missed Clear Register	Section 10.5.2.2.4
0318h	CCERR	EDMA3CC Error Register	Section 10.5.2.2.5
031Ch	CCERRCLR	EDMA3CC Error Clear Register	Section 10.5.2.2.6
0320h	EEVAL	Error Evaluate Register	Section 10.5.2.2.7
0340h	DRAE0	DMA Region Access Enable Register for Region 0	Section 10.5.2.3.1
0344h	DRAEH0	DMA Region Access Enable Register High for Region 0	Section 10.5.2.3.1
0348h	DRAE1	DMA Region Access Enable Register for Region 1	Section 10.5.2.3.1
034Ch	DRAEH1	DMA Region Access Enable Register High for Region 1	Section 10.5.2.3.1
0350h	DRAE2	DMA Region Access Enable Register for Region 2	Section 10.5.2.3.1
0354h	DRAEH2	DMA Region Access Enable Register High for Region 2	Section 10.5.2.3.1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 10-1. EDMACC Registers (continued)

Offset	Acronym	Register Description	Section
0358h	DRAE3	DMA Region Access Enable Register for Region 3	Section 10.5.2.3.1
035Ch	DRAEH3	DMA Region Access Enable Register High for Region 3	Section 10.5.2.3.1
0360h	DRAE4	DMA Region Access Enable Register for Region 4	Section 10.5.2.3.1
0364h	DRAEH4	DMA Region Access Enable Register High for Region 4	Section 10.5.2.3.1
0368h	DRAE5	DMA Region Access Enable Register for Region 5	Section 10.5.2.3.1
036Ch	DRAEH5	DMA Region Access Enable Register High for Region 5	Section 10.5.2.3.1
0370h	DRAE6	DMA Region Access Enable Register for Region 6	Section 10.5.2.3.1
0374h	DRAEH6	DMA Region Access Enable Register High for Region 6	Section 10.5.2.3.1
0378h	DRAE7	DMA Region Access Enable Register for Region 7	Section 10.5.2.3.1
037Ch	DRAEH7	DMA Region Access Enable Register High for Region 7	Section 10.5.2.3.1
0380h-039Ch	QRAE0-7	QDMA Region Access Enable Registers for Region 0-7	Section 10.5.2.3.2
0400h-04FCh	Q0E0-Q3E15	Event Queue Entry Registers Q0E0-Q3E15	Section 10.5.2.4.1
0600h-060Ch	QSTAT0-3	Queue Status Registers 0-3	Section 10.5.2.4.2
0620h	QWMTHRA	Queue Watermark Threshold A Register	Section 10.5.2.4.3
0640h	CCSTAT	EDMA3CC Status Register	Section 10.5.2.4.4
0800h	MPFAR	Memory Protection Fault Address Register	Section 10.5.2.5.1
0804h	MPFSR	Memory Protection Fault Status Register	Section 10.5.2.5.2
0808h	MPFCR	Memory Protection Fault Command Register	Section 10.5.2.5.3
080Ch	MPPAG	Memory Protection Page Attribut Register Global	Section 10.5.2.5.4
0810h-082Ch	MPPA0-7	Memory Protection Page Attribute Registers 0-7	Section 10.5.2.5.4
1000h	ER	Event Register	Section 10.5.2.6.1
1004h	ERH	Event Register High	Section 10.5.2.6.1
1008h	ECR	Event Clear Register	Section 10.5.2.6.2
100Ch	ECRH	Event Clear Register High	Section 10.5.2.6.2
1010h	ESR	Event Set Register	Section 10.5.2.6.3
1014h	ESRH	Event Set Register High	Section 10.5.2.6.3
1018h	CER	Chained Event Register	Section 10.5.2.6.4
101Ch	CERH	Chained Event Register High	Section 10.5.2.6.4
1020h	EER	Event Enable Register	Section 10.5.2.6.5
1024h	EERH	Event Enable Register High	Section 10.5.2.6.5
1028h	EECR	Event Enable Clear Register	Section 10.5.2.6.6
102Ch	EECRH	Event Enable Clear Register High	Section 10.5.2.6.6
1030h	EESR	Event Enable Set Register	Section 10.5.2.6.7
1034h	EESRH	Event Enable Set Register High	Section 10.5.2.6.7
1038h	SER	Secondary Event Register	Section 10.5.2.6.8
103Ch	SERH	Secondary Event Register High	Section 10.5.2.6.8
1040h	SECR	Secondary Event Clear Register	Section 10.5.2.6.9
1044h	SECRH	Secondary Event Clear Register High	Section 10.5.2.6.9
1050h	IER	Interrupt Enable Register	Section 10.5.2.7.1
1054h	IERH	Interrupt Enable Register High	Section 10.5.2.7.1
1058h	IECR	Interrupt Enable Clear Register	Section 10.5.2.7.2
105Ch	IECRH	Interrupt Enable Clear Register High	Section 10.5.2.7.2
1060h	IESR	Interrupt Enable Set Register	Section 10.5.2.7.3
1064h	IESRH	Interrupt Enable Set Register High	Section 10.5.2.7.3
1068h	IPR	Interrupt Pending Register	Section 10.5.2.7.4
106Ch	IPRH	Interrupt Pending Register High	Section 10.5.2.7.4
1070h	ICR	Interrupt Clear Register	Section 10.5.2.7.5

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 10-1. EDMACC Registers (continued)

Offset	Acronym	Register Description	Section
1074h	ICRH	Interrupt Clear Register High	Section 10.5.2.7.5
1078h	IEVAL	Interrupt Evaluate Register	Section 10.5.2.7.6
1080h	QER	QDMA Event Register	Section 10.5.2.8.1
1084h	QEER	QDMA Event Enable Register	Section 10.5.2.8.2
1088h	QEECR	QDMA Event Enable Clear Register	Section 10.5.2.8.3
108Ch	QEESR	QDMA Event Enable Set Register	Section 10.5.2.8.4
1090h	QSER	QDMA Secondary Event Register	Section 10.5.2.8.5
1094h	QSECR	QDMA Secondary Event Clear Register	Section 10.5.2.8.6
Shadow Region 0 Channel Registers			
2000h	ER	Event Register	
2004h	ERH	Event Register High	
2008h	ECR	Event Clear Register	
200Ch	ECRH	Event Clear Register High	
2010h	ESR	Event Set Register	
2014h	ESRH	Event Set Register High	
2018h	CER	Chained Event Register	
201Ch	CERH	Chained Event Register High	
2020h	EER	Event Enable Register	
2024h	EERH	Event Enable Register High	
2028h	EECR	Event Enable Clear Register	
202Ch	EECRH	Event Enable Clear Register High	
2030h	EESR	Event Enable Set Register	
2034h	EESRH	Event Enable Set Register High	
2038h	SER	Secondary Event Register	
203Ch	SERH	Secondary Event Register High	
2040h	SECR	Secondary Event Clear Register	
2044h	SECRH	Secondary Event Clear Register High	
2050h	IER	Interrupt Enable Register	
2054h	IERH	Interrupt Enable Register High	
2058h	IECR	Interrupt Enable Clear Register	
205Ch	IECRH	Interrupt Enable Clear Register High	
2060h	IESR	Interrupt Enable Set Register	
2064h	IESRH	Interrupt Enable Set Register High	
2068h	IPR	Interrupt Pending Register	
206Ch	IPRH	Interrupt Pending Register High	
2070h	ICR	Interrupt Clear Register	
2074h	ICRH	Interrupt Clear Register High	
2078h	IEVAL	Interrupt Evaluate Register	
2080h	QER	QDMA Event Register	
2084h	QEER	QDMA Event Enable Register	
2088h	QEECR	QDMA Event Enable Clear Register	
208Ch	QEESR	QDMA Event Enable Set Register	
2090h	QSER	QDMA Secondary Event Register	
2094h	QSECR	QDMA Secondary Event Clear Register	
2200h-2294h	-	Shadow Region 1 Channel Registers	
2400h-2494h	-	Shadow Region 2 Channel Registers	
...		...	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 10-1. EDMACC Registers (continued)

Offset	Acronym	Register Description	Section
2E00h-2E94h	-	Shadow Channel Registers for MP Space 7	

10.5.2.1 Global Registers

10.5.2.1.1 Peripheral Identification Register (PID)

The peripheral identification register (PID) uniquely identifies the EDMA3CC and the specific revision of the EDMA3CC.

The PID is shown in Figure 10-1 and described in Table 10-2.

Figure 10-1. Peripheral ID Register (PID)

31 – 16	15 – 0
PID	PID
R-4001h	R-1B00h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 10-2. Peripheral ID Register (PID) Field Descriptions

Bit	Field	Value	Description
31-0	PID	0-FFFF FFFFh	Peripheral identifier uniquely identifies the EDMA3CC and the specific revision of the EDMA3CC.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.1.2 EDMA3CC Configuration Register (CCCFG)

The EDMA3CC configuration register (CCCFG) provides the features/resources for the EDMA3CC in a particular device.

The CCCFG is shown in Figure 10-2 and described in Table 10-3.

Figure 10-2. EDMA3CC Configuration Register (CCCFG)

31–26	25	24
Reserved	MP_EXIST	CHMAP_EXIST
R-x	R-1	R-1

23–22	21–20	19	18–16
Reserved	NUM_REGN	Reserved	NUM_EVQUE
R-0	R-3h	R-x	R-3h

15	14–12	11	10–8
Reserved	NUM_PAENTRY	Reserved	NUM_INTCH
R-x	R-5h	R-x	R-4h

7	6–4	3	2–0
Reserved	NUM_QDMACH	Reserved	NUM_DMACH
R-x	R-4h	R-x	R-5h

LEGEND: R = Read only; -*n* = value after reset; -x = value is indeterminate after reset

Table 10-3. EDMA3CC Configuration Register (CCCFG) Field Descriptions

Bit	Field	Value	Description
31-26	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
25	MP_EXIST		Memory protection existence.
		0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
		1	Memory protection logic included.
24	CHMAP_EXIST		Channel mapping existence.
		0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
		1	Channel mapping logic included.
23-22	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
21-20	NUM_REGN	0-3h	Number of MP and shadow regions.
		0-2h	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
		3h	8 regions.
19	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
18-16	NUM_EVQUE	0-7h	Number of queues/number of TCs.
		0-2h	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
		3h	4 EDMA3TCs/Event Queues.
		4h-7h	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
15	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 10-3. EDMA3CC Configuration Register (CCCFG) Field Descriptions (continued)

Bit	Field	Value	Description
14-12	NUM_PAENTRY	0-7h	Number of PaRAM sets.
		0-3h	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
		5h	512 PaRAM sets.
		5h-7h	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
11	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
10-8	NUM_INTCH	0-7h	Number of interrupt channels.
		0-3h	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
		4h	64 interrupt channels.
		5h-7h	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
7	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
6-4	NUM_QDMACH	0-7h	Number of QDMA channels.
		0-1h	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
		4h	8 QDMA channels.
		3h-7h	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
3	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
2-0	NUM_DMACH	0-7h	Number of DMA channels.
		0-4h	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
		5h	64 DMA channels.
		6h-7h	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.1.3 DMA Channel Map *n* Registers (DCHMAP*n*)

The DMA channel map *n* register (DCHMAP*n*) is shown in Figure 10-3 and described in Table 10-4.

Figure 10-3. DMA Channel Map *n* Registers (DCHMAP*n*)

31															16
Reserved															
R-0															

15	14	13												5	4									0
Reserved		PAENTRY													Reserved									
R-0		R/W-0													R-0									

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 10-4. DMA Channel Map *n* Registers (DCHMAP*n*) Field Descriptions

Bit	Field	Value	Description
31-14	Reserved	0	Reserved
13-5	PAENTRY	0-1FFh	Points to the PaRAM set number for DMA channel *n*.
4-0	Reserved	0	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.1.4 QDMA Channel Map *n* Registers (QCHMAP*n*)

Each QDMA channel in EDMA3CC can be associated with any PaRAM set available on the device. Furthermore, the specific trigger word (0-7) of the PaRAM set can be programmed. The PaRAM set association and trigger word for every QDMA channel register is configurable using the QDMA channel map *n* register (QCHMAP*n*).

The QCHMAP*n* is shown in Figure 10-4 and described in Table 10-5.

NOTE: At reset the QDMA channel map registers for all QDMA channels point to PaRAM set 0. If an application makes use of both a DMA channel that points to PaRAM set 0 and any QDMA channels, ensure that QCHMAP*n* is programmed appropriately to point to a different PaRAM entry.

Figure 10-4. QDMA Channel Map *n* Registers (QCHMAP*n*)

31–16	15–14	13–5	4–2	1–0
Reserved	Reserved	PAENTRY	TRWORD	Reserved
R-0	R-0	R/W-0	R/W-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 10-5. QDMA Channel Map *n* Registers (QCHMAP*n*) Field Descriptions

Bit	Field	Value	Description
31-14	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
13-5	PAENTRY	0-1FFh	PAENTRY points to the PaRAM set number for QDMA channel *n*.
		0-FFh	Parameter entry 0 through 511.
		100h-1FFh	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
4-2	TRWORD	0-7h	Points to the specific trigger word of the PaRAM set defined by PAENTRY. A write to the trigger word results in a QDMA event being recognized.
1-0	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.1.5 DMA Channel Queue *n* Number Registers (DMAQNUM*n*)

The DMA channel queue number register (DMAQNUM*n*) allows programmability of each of the 64 DMA channels in the EDMA3CC to submit its associated synchronization event to any event queue in the EDMA3CC. At reset, all channels point to event queue 0.

The DMAQNUM*n* is shown in Figure 10-5 and described in Table 10-6. Table 10-7 shows the channels and their corresponding bits in DMAQNUM*n*.

NOTE: Because the event queues in EDMA3CC have a fixed association to the transfer controllers, that is, Q0 TRs are submitted to TC0, Q1 TRs are submitted to TC1, etc., by programming DMAQNUM*n* for a particular DMA channel *n* also dictates which transfer controller is utilized for the data movement (or which EDMA3TC receives the TR request).

Figure 10-5. DMA Channel Queue *n* Number Registers (DMAQNUM*n*)

31	30 28	27	26 24	23	22 20	19	18 16
Rsvd	E*n*	Rsvd	E*n*	Rsvd	E*n*	Rsvd	E*n*
R-0	R/W-0	R-0	R/W-0	R-0	R/W-0	R-0	R/W-0

15	14 12	11	10 8	7	6 4	3	2 0
Rsvd	E*n*	Rsvd	E*n*	Rsvd	E*n*	Rsvd	E*n*
R-0	R/W-0	R-0	R/W-0	R-0	R/W-0	R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 10-6. DMA Channel Queue *n* Number Registers (DMAQNUM*n*) Field Descriptions

Bit	Field	Value	Description
31-0	E*n*	0-7h	DMA queue number. Contains the event queue number to be used for the corresponding DMA channel. Programming DMAQNUM*n* for an event queue number to a value more then the number of queues available in the EDMA3CC results in undefined behavior.
		0	Event *n* is queued on Q0.
		1h	Event *n* is queued on Q1.
		2h	Event *n* is queued on Q2.
		3h	Event *n* is queued on Q3.
		4h-7h	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.

Table 10-7. Bits in DMAQNUM*n*

	Channel Number (DMAQNUM*n*)							
E*n* bit	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0-2	E0	E8	E16	E24	E32	E40	E48	E56
4-6	E1	E9	E17	E25	E33	E41	E49	E57
8-10	E2	E10	E18	E26	E34	E42	E50	E58
12-14	E3	E11	E19	E27	E35	E43	E51	E59
16-18	E4	E12	E20	E28	E36	E44	E52	E60
20-22	E5	E13	E21	E29	E37	E45	E53	E61
24-26	E6	E14	E22	E30	E38	E46	E54	E62
28-30	E7	E15	E23	E31	E39	E47	E55	E63

10.5.2.1.6 QDMA Channel Queue Number Register (QDMAQNUM)

The QDMA channel queue number register (QDMAQNUM) is used to program all the QDMA channels in the EDMA3CC to submit the associated QDMA event to any of the event queues in the EDMA3CC.

The QDMAQNUM is shown in Figure 10-6 and described in Table 10-8.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 10-6. QDMA Channel Queue Number Register (QDMAQNUM)

31	30–28	27	26–24	23	22–20	19	18–16
Rsvd	E7	Rsvd	E6	Rsvd	E5	Rsvd	E4
R-0	R/W-0	R-0	R/W-0	R-0	R/W-0	R-0	R/W-0

15	14–12	11	10–8	7	6–4	3	2–0
Rsvd	E3	Rsvd	E2	Rsvd	E1	Rsvd	E0
R-0	R/W-0	R-0	R/W-0	R-0	R/W-0	R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 10-8. QDMA Channel Queue Number Register (QDMAQNUM) Field Descriptions

Bit	Field	Value	Description
31-15	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
14-0	E*n*	0-7h	QDMA queue number. Contains the event queue number to be used for the corresponding QDMA channel.
		0	Event *n* is queued on Q0.
		1h	Event *n* is queued on Q1.
		2h	Event *n* is queued on Q2.
		3h	Event *n* is queued on Q3.
		4h-7h	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.1.7 Queue Priority Register (QUEPRI)

The queue priority register (QUEPRI) allows you to change the priority of the individual queues and the priority of the transfer request (TR) associated with the events queued in the queue. Because the queue to EDMA3TC mapping is fixed, programming QUEPRI essentially governs the priority of the associated transfer controller(s) read/write commands with respect to the other bus masters in the device. You can modify the EDMA3TC priority to obtain the desired system performance.

The QUEPRI is shown in Figure 10-7 and described in Table 10-9.

Figure 10-7. Queue Priority Register (QUEPRI)

31–16	15	14–12	11	10–8	7	6–4	3	2–0
Reserved	Rsvd	PRIQ3	Rsvd	PRIQ2	Rsvd	PRIQ1	Rsvd	PRIQ0
R-0	R-0	R/W-0	R-0	R/W-0		R/W-0	R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 10-9. Queue Priority Register (QUEPRI) Field Descriptions

Bit	Field	Value	Description
31-15	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
14-12	PRIQ3	0-7h	Priority level for queue 3. Dictates the priority level used by TC3 relative to other masters in the device. A value of 0 means highest priority and a value of 7 means lowest priority.
11	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
10-8	PRIQ2	0-7h	Priority level for queue 2. Dictates the priority level used by TC2 relative to other masters in the device. A value of 0 means highest priority and a value of 7 means lowest priority.
7	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
6-4	PRIQ1	0-7h	Priority level for queue 1. Dictates the priority level used by TC1 relative to other masters in the device. A value of 0 means highest priority and a value of 7 means lowest priority.
3	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
2-0	PRIQ0	0-7h	Priority level for queue 0. Dictates the priority level used by TC0 relative to other masters in the device. A value of 0 means highest priority and a value of 7 means lowest priority.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.2 Error Registers

The EDMA3CC contains a set of registers that provide information on missed DMA and/or QDMA events, and instances when event queue thresholds are exceeded. If any of the bits in these registers is set, it results in the EDMA3CC generating an error interrupt.

10.5.2.2.1 Event Missed Registers (EMR/EMRH)

For a particular DMA channel, if a second event is received prior to the first event getting cleared/serviced, the bit corresponding to that channel is set/asserted in the event missed registers (EMR/EMRH). All trigger types are treated individually, that is, manual triggered (ESR/ESRH), chain triggered (CER/CERH), and event triggered (ER/ERH) are all treated separately. The EMR/EMRH bits for a channel are also set if an event on that channel encounters a NULL entry (or a NULL TR is serviced). If any EMR/EMRH bit is set (and all errors, including bits in other error registers (QEMR, CCERR) were previously cleared), the EDMA3CC generates an error interrupt. For details on EDMA3CC error interrupt generation, see *Error Interrupts*.

The EMR is shown in Figure 10-8 and described in Table 10-10. The EMRH is shown in Figure 10-9 and described in Table 10-11.

Figure 10-8. Event Missed Register (EMR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 10-10. Event Missed Register (EMR) Field Descriptions

Bit	Field	Value	Description
31-0	E*n*		Channel 0-31 event missed. E*n* is cleared by writing a 1 to the corresponding bit in the event missed clear register (EMCR).
		0	No missed event.
		1	Missed event occurred.

Figure 10-9. Event Missed Register High (EMRH)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 10-11. Event Missed Register High (EMRH) Field Descriptions

Bit	Field	Value	Description
31-0	E*n*		Channel 32–63 event missed. E*n* is cleared by writing a 1 to the corresponding bit in the event missed clear register high (EMCRH).
		0	No missed event.
		1	Missed event occurred.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.2.2 Event Missed Clear Registers (EMCR/EMCRH)

Once a missed event is posted in the event missed registers (EMR/EMRH), the bit remains set and you need to clear the set bit(s). This is done by way of CPU writes to the event missed clear registers (EMCR/EMCRH). Writing a 1 to any of the bits clears the corresponding missed event (bit) in EMR/EMRH; writing a 0 has no effect.

The EMCR is shown in Figure 10-10 and described in Table 10-12. The EMCRH is shown in Figure 10-11 and described in Table 10-13.

Figure 10-10. Event Missed Clear Register (EMCR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0

LEGEND: W = Write only; -*n* = value after reset

Table 10-12. Event Missed Clear Register (EMCR) Field Descriptions

Bit	Field	Value	Description
31-0	E*n*		Event missed 0-31 clear. All error bits must be cleared before additional error interrupts will be asserted by the EDMA3CC.
		0	No effect.
		1	Corresponding missed event bit in the event missed register (EMR) is cleared (E*n* = 0).

Figure 10-11. Event Missed Clear Register High (EMCRH)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0

LEGEND: W = Write only; -*n* = value after reset

Table 10-13. Event Missed Clear Register High (EMCRH) Field Descriptions

Bit	Field	Value	Description
31-0	E*n*		Event missed 32–63 clear. All error bits must be cleared before additional error interrupts will be asserted by the EDMA3CC.
		0	No effect.
		1	Corresponding missed event bit in the event missed register high (EMRH) is cleared (E*n* = 0).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.2.3 QDMA Event Missed Register (QEMR)

For a particular QDMA channel, if two QDMA events are detected without the first event getting cleared/serviced, the bit corresponding to that channel is set/asserted in the QDMA event missed register (QEMR). The QEMR bits for a channel are also set if a QDMA event on the channel encounters a NULL entry (or a NULL TR is serviced). If any QEMR bit is set (and all errors, including bits in other error registers (EMR/EMRH, CCERR) were previously cleared), the EDMA3CC generates an error interrupt. For details on EDMA3CC error interrupt generation, see *Error Interrupts*.

The QEMR is shown in Figure 10-12 and described in Table 10-14.

Figure 10-12. QDMA Event Missed Register (QEMR)

31							16
Reserved							
R-0							

15	7	6	5	4	3	2	1	0
Reserved	E7	E6	E5	E4	E3	E2	E1	E0
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 10-14. QDMA Event Missed Register (QEMR) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
7-0	E*n*		Channel 0-7 QDMA event missed. E*n* is cleared by writing a 1 to the corresponding bit in the QDMA event missed clear register (QEMCR).
		0	No missed event.
		1	Missed event occurred.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.2.4 QDMA Event Missed Clear Register (QEMCR)

Once a missed event is posted in the QDMA event missed registers (QEMR), the bit remains set and you need to clear the set bit(s). This is done by way of CPU writes to the QDMA event missed clear registers (QEMCR). Writing a 1 to any of the bits clears the corresponding missed event (bit) in QEMR; writing a 0 has no effect.

The QEMCR is shown in Figure 10-13 and described in Table 10-15.

Figure 10-13. QDMA Event Missed Clear Register (QEMCR)

31–16	15–8	7	6	5	4	3	2	1	0
Reserved	Reserved	E7	E6	E5	E4	E3	E2	E1	E0
R-0	R-0	w-0	w-0	w-0	w-0	W-0	W-0	W-0	W-0

LEGEND: W = Write only; -*n* = value after reset

Table 10-15. QDMA Event Missed Clear Register (QEMCR) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
7-0	E*n*		QDMA event missed clear. All error bits must be cleared before additional error interrupts will be asserted by the EDMA3CC.
		0	No effect.
		1	Corresponding missed event bit in the QDMA event missed register (QEMR) is cleared (E*n* = 0).

10.5.2.2.5 EDMA3CC Error Register (CCERR)

The EDMA3CC error register (CCERR) indicates whether or not at any instant of time the number of events queued up in any of the event queues exceeds or equals the threshold/watermark value that is set in the queue watermark threshold register (QWMTHRA). Additionally, CCERR also indicates if when the number of outstanding TRs that have been programmed to return transfer completion code (TRs which have the TCINTEN or TCCHEN bit in OPT set) to the EDMA3CC has exceeded the maximum allowed value of 63. If any bit in CCERR is set (and all errors, including bits in other error registers (EMR/EMRH, QEMR) were previously cleared), the EDMA3CC generates an error interrupt. For details on EDMA3CC error interrupt generation, see *Error Interrupts*. Once the error bits are set in CCERR, they can only be cleared by writing to the corresponding bits in the EDMA3CC error clear register (CCERRCLR).

The CCERR is shown in Figure 10-14 and described in Table 10-16.

Figure 10-14. EDMA3CC Error Register (CCERR)

31–17	16	15–4	3	2	1	0
Reserved	TCCERR	Reserved	QTHRXCD3	QTHRXCD2	QTHRXCD1	QTHRXCD0
R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 10-16. EDMA3CC Error Register (CCERR) Field Descriptions

Bit	Field	Value	Description
31-17	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
16	TCCERR		Transfer completion code error. TCCERR is cleared by writing a 1 to the corresponding bit in the EDMA3CC error clear register (CCERRCLR).
		0	Total number of allowed TCCs outstanding has not been reached.
		1	Total number of allowed TCCs has been reached.
15-4	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
3	QTHRXCD3		Queue threshold error for queue 3. QTHRXCD3 is cleared by writing a 1 to the corresponding bit in the EDMA3CC error clear register (CCERRCLR).
		0	Watermark/threshold has not been exceeded.
		1	Watermark/threshold has been exceeded.
2	QTHRXCD2		Queue threshold error for queue 2. QTHRXCD2 is cleared by writing a 1 to the corresponding bit in the EDMA3CC error clear register (CCERRCLR).
		0	Watermark/threshold has not been exceeded.
		1	Watermark/threshold has been exceeded.
1	QTHRXCD1		Queue threshold error for queue 1 . QTHRXCD1 is cleared by writing a 1 to the corresponding bit in the EDMA3CC error clear register (CCERRCLR).
		0	Watermark/threshold has not been exceeded.
		1	Watermark/threshold has been exceeded.
0	QTHRXCD0		Queue threshold error for queue 0. QTHRXCD0 is cleared by writing a 1 to the corresponding bit in the EDMA3CC error clear register (CCERRCLR).
		0	Watermark/threshold has not been exceeded.
		1	Watermark/threshold has been exceeded.

10.5.2.2.6 EDMA3CC Error Clear Register (CCERRCLR)

The EDMA3CC error clear register (CCERRCLR) is used to clear any error bits that are set in the EDMA3CC error register (CCERR). In addition, CCERRCLR also clears the values of some bit fields in the queue status registers (QSTAT*n*) associated with a particular event queue. Writing a 1 to any of the bits clears the corresponding bit in CCERR; writing a 0 has no effect.

The CCERRCLR is shown in Figure 10-15 and described in Table 10-17.

Figure 10-15. EDMA3CC Error Clear Register (CCERRCLR)

31	17	16
Reserved		TCCERR
W-0		W-0

15	4	3	2	1	0
Reserved		QTHRXCD3	QTHRXCD2	QTHRXCD1	QTHRXCD0
W-0		W-0	W-0	W-0	W-0

LEGEND: W = Write only; -*n* = value after reset

Table 10-17. EDMA3CC Error Clear Register (CCERRCLR) Field Descriptions

Bit	Field	Value	Description
31-17	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
16	TCCERR		Transfer completion code error clear.
		0	No effect.
		1	Clears the TCCERR bit in the EDMA3CC error register (CCERR).

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Table 10-17. EDMA3CC Error Clear Register (CCERRCLR) Field Descriptions (continued)

Bit	Field	Value	Description
15-4	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
3	QTHRXCD3		Queue threshold error clear for queue 3.
		0	No effect.
		1	Clears the QTHRXCD3 bit in the EDMA3CC error register (CCERR) and the WM and THRXCD bits in the queue status register 3 (QSTAT3).
2	QTHRXCD2		Queue threshold error clear for queue 2.
		0	No effect.
		1	Clears the QTHRXCD2 bit in the EDMA3CC error register (CCERR) and the WM and THRXCD bits in the queue status register 2 (QSTAT2).
1	QTHRXCD1		Queue threshold error clear for queue 1.
		0	No effect.
		1	Clears the QTHRXCD1 bit in the EDMA3CC error register (CCERR) and the WM and THRXCD bits in the queue status register 1 (QSTAT1).
0	QTHRXCD0		Queue threshold error clear for queue 0.
		0	No effect.
		1	Clears the QTHRXCD0 bit in the EDMA3CC error register (CCERR) and the WM and THRXCD bits in the queue status register 0 (QSTAT0).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.2.7 Error Evaluation Register (EEVAL)

The EDMA3CC error interrupt is asserted whenever an error bit is set in any of the error registers (EMR/EMRH, QEMR, and CCERR). For subsequent error bits that get set, the EDMA3CC error interrupt is reasserted only when transitioning from an “all the error bits cleared” to “at least one error bit is set”. Alternatively, a CPU write of 1 to the EVAL bit in the error evaluation register (EEVAL) results in reasserting the EDMA3CC error interrupt, if there are any outstanding error bits set due to subsequent error conditions. Writes of 0 have no effect.

The EEVAL is shown in Figure 10-16 and described in Table 10-18.

Figure 10-16. Error Evaluation Register (EEVAL)

31			16
Reserved			
R-0			

15	2	1	0
Reserved		Rsvd	EVAL
R-0		R/W-0	W-0

LEGEND: R/W = Read/Write; R = Read only; W = Write only; -*n* = value after reset

Table 10-18. Error Evaluation Register (EEVAL) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
1	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
0	EVAL		Error interrupt evaluate.
		0	No effect.
		1	EDMA3CC error interrupt will be pulsed if any errors have not been cleared in any of the error registers (EMR/EMRH, QEMR, or CCERR).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.3 Region Access Enable Registers

The region access enable register group consists of the DMA access enable registers (DRAE*m* and DRAEH*m*) and the QDMA access enable registers (QRAE*m*). Where *m* is the number of shadow regions in the EDMA3CC memory map for a device. You can configure these registers to assign ownership of DMA/QDMA channels to a particular shadow region.

10.5.2.3.1 DMA Region Access Enable for Region m (DRAEm)

The DMA region access enable register for shadow region *m* (DRAE*m*/DRAEH*m*) is programmed to allow or disallow read/write accesses on a bit-by-bit bases for all DMA registers in the shadow region *m* view of the DMA channel registers. See the EDMA3CC register memory map (Table 10-1) for a list of all the DMA channel and interrupt registers mapped in the shadow region view. Additionally, the DRAE*m*/DRAEH*m* configuration determines completion of which DMA channels will result in assertion of the shadow region *m* DMA completion interrupt (see *EDMA3 Interrupts*).

The DRAE*m* is shown in Figure 10-17 and described in Table 10-19. The DRAEH*m* is shown in Figure 10-18 and described in Table 10-19.

Figure 10-17. DMA Region Access Enable Register for Region *m* (DRAE*m*)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

LEGEND: R/W = Read/Write; -*n* = value after reset

Figure 10-18. DMA Region Access Enable High Register for Region *m* (DRAEH*m*)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 10-19. DMA Region Access Enable Registers for Region M (DRAE*m*/DRAEH*m*) Field Descriptions

Bit	Field	Value	Description
31-0	E*n*		DMA region access enable for bit *n*/channel *n* in region *m*.
		0	Accesses via region *m* address space to bit *n* in any DMA channel register are not allowed. Reads return 0 on bit *n* and writes do not modify the state of bit *n*. Enabled interrupt bits for bit *n* do not contribute to the generation of a transfer completion interrupt for shadow region *m*.
		1	Accesses via region *m* address space to bit *n* in any DMA channel register are allowed. Reads return the value from bit *n* and writes modify the state of bit *n*. Enabled interrupt bits for bit *n* contribute to the generation of a transfer completion interrupt for shadow region *m*.

10.5.2.3.2 QDMA Region Access Enable Registers (QRAEm)

The QDMA region access enable register for shadow region *m* (QRAE*m*) is programmed to allow or disallow read/write accesses on a bit-by-bit bases for all QDMA registers in the shadow region *m* view of the QDMA registers. This includes all 4-bit QDMA registers.

The QRAE*m* is shown in Figure 10-19 and described in Table 10-20.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 10-19. QDMA Region Access Enable for Region *m* (QRAE*m*)32-bit, 2 Rows

31								16
Reserved								
R-0								

15	8	7	6	5	4	3	2	1	0
Reserved		E7	E6	E5	E4	E3	E2	E1	E0
R-0		RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 10-20. QDMA Region Access Enable for Region M (QRAE*m*) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
7-0	E*n*		QDMA region access enable for bit *n*/QDMA channel *n* in region *m*.
		0	Accesses via region *m* address space to bit *n* in any QDMA channel register are not allowed. Reads return 0 on bit *n* and writes do not modify the state of bit *n*.
		1	Accesses via region *m* address space to bit *n* in any QDMA channel register are allowed. Reads return the value from bit *n* and writes modify the state of bit *n*.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.4 Status/Debug Visibility Registers

The following set of registers provide visibility into the event queues and a TR life cycle. These are useful for system debug as they provide in-depth visibility for the events queued up in the event queue and also provide information on what parts of the EDMA3CC logic are active once the event has been received by the EDMA3CC.

10.5.2.4.1 Event Queue Entry Registers (QxEy)

The event queue entry registers (Q*x*E*y*) exist for all 16 queue entries (the maximum allowed queue entries) for all event queues in the EDMA3CC.

There are Q0E0 to Q0E15, Q1E0 to Q1E15, Q2E0 to Q2E15, and Q3E0 to Q3E15. Each register details the event number (ENUM) and the event type (ETYPE). For example, if the value in Q1E4 is read as 000 004Fh, this means the 4th entry in queue 1 is a manually-triggered event on DMA channel 15.

The Q*x*E*y* is shown in Figure 10-20 and described in Table 10-21.

Figure 10-20. Event Queue Entry Registers (Q*x*E*y*)

31–16	15–8	7–6	5–0
Reserved	Reserved	ETYPE	ENUM
R-0	R-0	R-x	R-x

LEGEND: R = Read only; -*n* = value after reset; -x = value is indeterminate after reset

Table 10-21. Event Queue Entry Registers (Q*x*E*y*) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
7-6	ETYPE	0-3h	Event entry *y* in queue *x*. Specifies the specific event type for the given entry in the event queue.
		0	Event triggered via ER.
		1h	Manual triggered via ESR.
		2h	Chain triggered via CER.
		3h	Auto-triggered via QER.
5-0	ENUM	0-3Fh	Event entry *y* in queue *x*. Event number:
		0-3h	QDMA channel number (0 to 3).
		0-3Fh	DMA channel/event number (0 to 63).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.4.2 Queue Status Registers (QSTAT*n*)

The queue status registers (QSTAT*n*) is shown in Figure 10-21 and described in Table 10-22.

Figure 10-21. Queue Status Register *n* (QSTAT*n*)

31–25	24	23–21	20–16
Reserved	THRXCD	Reserved	WM
R-0	R-0	R-0	R-0

15–13	12–8	7–4	3–0
Reserved	NUMVAL	Reserved	STRTPTR
R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 10-22. Queue Status Register *n* (QSTAT*n*) Field Descriptions

Bit	Field	Value	Description
31-25	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
24	THRXCD		Threshold exceeded. THRXCD is cleared by writing a 1 to the corresponding QTHRXCD*n* bit in the EDMA3CC error clear register (CCERRCLR).
		0	Threshold specified by the Q*n* bit in the queue watermark threshold A register (QWMTHRA) has not been exceeded.
		1	Threshold specified by the Q*n* bit in the queue watermark threshold A register (QWMTHRA) has been exceeded.
23-21	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
20-16	WM	0-10h	Watermark for maximum queue usage. Watermark tracks the most entries that have been in queue *n* since reset or since the last time that the watermark (WM) bit was cleared. WM is cleared by writing a 1 to the corresponding QTHRXCD*n* bit in the EDMA3CC error clear register (CCERRCLR).
		0-10h	Legal values are 0 (empty) to 10h (full).
		11h-1Fh	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
15-13	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
12-8	NUMVAL	0-10h	Number of valid entries in queue *n*. The total number of entries residing in the queue manager FIFO at a given instant. Always enabled.
		0-10h	Legal values are 0 (empty) to 10h (full).
		11h-1Fh	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
7-4	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
3-0	STRTPTR	0-Fh	Start pointer. The offset to the head entry of queue *n*, in units of entries. Always enabled. Legal values are 0 (0th entry) to Fh (15th entry).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.4.3 Queue Watermark Threshold A Register (QWMTHRA)

The queue watermark threshold A register (QWMTHRA) is shown in Figure 10-22 and described in Table 10-23.

Figure 10-22. Queue Watermark Threshold A Register (QWMTHRA)

31–29	28–24	23–21	20–16
Reserved	Q3	Reserved	Q2
R-0	R/W-10	R-0	R/W-10

15–13	12–8	7–5	4–0
Reserved	Q1	Reserved	Q0
R-0	R/W-10h	R-0	R/W-10h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 10-23. Queue Watermark Threshold A Register (QWMTHRA) Field Descriptions

Bit	Field	Value	Description
31-29	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
28-24	Q3	0-1Fh	Queue threshold for queue 3 value. The QTHRXCD3 bit in the EDMA3CC error register (CCERR) and the THRXCD bit in the queue status register 3 (QSTAT3) are set when the number of events in queue 3 at an instant in time (visible via the NUMVAL bit in QSTAT3) equals or exceeds the value specified by Q3.
		0-10h	The default is 16 (maximum allowed).
		11h	Disables the threshold errors.
		12h-1Fh	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
23-21	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
20-16	Q2	0-1Fh	Queue threshold for queue 2 value. The QTHRXCD2 bit in the EDMA3CC error register (CCERR) and the THRXCD bit in the queue status register 2 (QSTAT2) are set when the number of events in queue 2 at an instant in time (visible via the NUMVAL bit in QSTAT2) equals or exceeds the value specified by Q2.
		0-10h	The default is 16 (maximum allowed).
		11h	Disables the threshold errors.
		12h-1Fh	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
15-13	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
12-8	Q1		Queue threshold for queue 1 value. The QTHRXCD1 bit in the EDMA3CC error register (CCERR) and the THRXCD bit in the queue status register 1 (QSTAT1) are set when the number of events in queue 1 at an instant in time (visible via the NUMVAL bit in QSTAT1) equals or exceeds the value specified by Q1.
		0-10h	The default is 16 (maximum allowed).
		11h	Disables the threshold errors.
		12h-1Fh	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
7-5	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
4-0	Q0		Queue threshold for queue 0 value. The QTHRXCD0 bit in the EDMA3CC error register (CCERR) and the THRXCD bit in the queue status register 0 (QSTAT0) are set when the number of events in queue 0 at an instant in time (visible via the NUMVAL bit in QSTAT0) equals or exceeds the value specified by Q0.
		0-10h	The default is 16 (maximum allowed).
		11h	Disables the threshold errors.
		12h-1Fh	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.4.4 EDMA3CC Status Register (CCSTAT)

The EDMA3CC status register (CCSTAT) has a number of status bits that reflect which parts of the EDMA3CC logic is active at any given instant of time. The CCSTAT is shown in Figure 10-23 and described in Table 10-24.

Figure 10-23. EDMA3CC Status Register (CCSTAT)

31							24
Reserved							
R-0							
23	22	21	20	19	18	17	16
Reserved	Reserved			QUEACTV3	QUEACTV2	QUEACTV1	QUEACTV0
R-0	R-0			R-0	R-0	R-0	R-0
15	14	13	12	11	10	9	8
Reserved		COMPACTV					
R-0		R-0					
7	6	5	4	3	2	1	0
Reserved			ACTV	Reserved	TRACTV	QEVTACTV	EVTACTV
R-0			R-0	R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 10-24. EDMA3CC Status Register (CCSTAT) Field Descriptions

Bit	Field	Value	Description
31-20	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
19	QUEACTV3		Queue 3 active.
		0	No events are queued in queue 3.
		1	At least one TR is queued in queue 3.
18	QUEACTV2		Queue 2 active.
		0	No events are queued in queue 2.
		1	At least one TR is queued in queue 2.
17	QUEACTV1		Queue 1 active.
		0	No events are queued in queue 1.
		1	At least one TR is queued in queue 1.
16	QUEACTV0		Queue 0 active.
		0	No events are queued in queue 0.
		1	At least one TR is queued in queue 0.
15-14	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
13-8	COMPACTV	0-3Fh	Completion request active. The COMPACTV field reflects the count for the number of completion requests submitted to the transfer controllers. This count increments every time a TR is submitted and is programmed to report completion (the TCINTEN or TCCCHEN bits in OPT in the parameter entry associated with the TR are set). The counter decrements for every valid TCC received back from the transfer controllers. If at any time the count reaches a value of 63, the EDMA3CC will not service any new TRs until the count is less then 63 (or return a transfer completion code from a transfer controller, which would decrement the count).
		0	No completion requests outstanding.
		1h-3Fh	Total of 1 completion request to 63 completion requests are outstanding.
7-5	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 10-24. EDMA3CC Status Register (CCSTAT) Field Descriptions (continued)

Bit	Field	Value	Description
4	ACTV		Channel controller active. Channel controller active is a logical-OR of each of the *ACTV bits. The ACTV bit remains high through the life of a TR.
		0	Channel is idle..
		1	Channel is busy.
3	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
2	TRACTV		Transfer request active.
		0	Transfer request processing/submission logic is inactive.
		1	Transfer request processing/submission logic is active.
1	QEVTACTV		QDMA event active.
		0	No enabled QDMA events are active within the EDMA3CC.
		1	At least one enabled QDMA event (QER) is active within the EDMA3CC.
0	EVTACTV		DMA event active.
		0	No enabled DMA events are active within the EDMA3CC.
		1	At least one enabled DMA event (ER and EER, ESR, CER) is active within the EDMA3CC.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.5 Memory Protection Address Space

10.5.2.5.1 Memory Protection Fault Address Register (MPFAR)

The memory protection fault address register (MPFAR) is shown in Figure 10-24 and described in Table 10-25. A CPU write of 1 to the MPFCLR bit in the memory protection fault command register (MPFCR) causes any error conditions stored in MPFAR to be cleared.

Figure 10-24. Memory Protection Fault Address Register (MPFAR)

31															16
FADDR															
R-0															

15															0
FADDR															
R-0															

LEGEND: R = Read only; -*n* = value after reset

Table 10-25. Memory Protection Fault Address Register (MPFAR) Field Descriptions

Bit	Field	Value	Description
31-0	FADDR	0-FFFF FFFFh	Fault address. This 32-bit read-only status register contains the fault address when a memory protection violation is detected. This register can only be cleared via the memory protection fault command register (MPFCR).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.5.2 Memory Protection Fault Status Register (MPFSR)

The memory protection fault status register (MPFSR) is shown in Figure 10-25 and described in Table 10-26. A CPU write of 1 to the MPFCLR bit in the memory protection fault command register (MPFCR) causes any error conditions stored in MPFSR to be cleared.

Figure 10-25. Memory Protection Fault Status Register (MPFSR)

31															16
Reserved															
R-0															

15	13	12	9	8	6	5	4	3	2	1	0
Reserved		FID		Reserved		SRE	SWE	SXE	URE	UWE	UXE
R-0		R-0		R-0		R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 10-26. Memory Protection Fault Status Register (MPFSR) Field Descriptions

Bit	Field	Value	Description
31-13	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
12-9	FID	0-Fh	Faulted identification. FID contains valid information if any of the MP error bits (UXE, UWE, URE, SXE, SWE, SRE) are nonzero (that is, if an error has been detected.) The FID field contains the privilege ID for the specific request/requestor that resulted in an MP error.
8-6	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
5	SRE		Supervisor read error.
		0	No error detected.
		1	Supervisor level task attempted to read from a MP page without SR permissions.
4	SWE		Supervisor write error.
		0	No error detected.
		1	Supervisor level task attempted to write to a MP page without SW permissions.
3	SXE		Supervisor execute error.
		0	No error detected.
		1	Supervisor level task attempted to execute from a MP page without SX permissions.
2	URE		User read error.
		0	No error detected.
		1	User level task attempted to read from a MP page without UR permissions.
1	UWE		User write error.
		0	No error detected.
		1	User level task attempted to write to a MP page without UW permissions.
0	UXE		User execute error.
		0	No error detected.
		1	User level task attempted to execute from a MP page without UX permissions.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.5.3 Memory Protection Fault Command Register (MPFCR)

The memory protection fault command register (MPFCR) is shown in Figure 10-26 and described in Table 10-27.

Figure 10-26. Memory Protection Fault Command Register (MPFCR)

31	16
Reserved	
R-0	

15	1	0
Reserved		MPFCLR
R-0		W-0

LEGEND: R = Read only; W = Write only; -*n* = value after reset

Table 10-27. Memory Protection Fault Command Register (MPFCR) Field Descriptions

Bit	Field	Value	Description
31-1	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
0	MPFCLR		Fault clear register.
		0	CPU write of 0 has no effect.
		1	CPU write of 1 to the MPFCLR bit causes any error conditions stored in the memory protection fault address register (MPFAR) and the memory protection fault status register (MPFSR) to be cleared.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.5.4 Memory Protection Page Attribute Register (MPPA*n*)

The memory protection page attribute register (MPPA*n*) is shown in Figure 10-27 and described in Table 10-28.

Figure 10-27. Memory Protection Page Attribute Register (MPPA*n*)

31–16															
Reserved															
R-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AID5	AID4	AID3	AID2	AID1	AID0	EXT	Rsvd	Reserved		SR	SW	SX	UR	UW	UX
RW-1	RW-1	RW-1	RW-1	RW-1	RW-1	RW-1	R-0	RW-1		RW-1	RW-1	RW-0	RW-1	RW-1	RW-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 10-28. Memory Protection Page Attribute Register (MPPA*n*) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
15-10	AID*m*		Allowed ID 'N'
		0	Requests with Privilege ID == N are not allowed to region M, regardless of permission settings (UW, UR, SW, SR).
		1	Requests with Privilege ID == N are permitted, if access type is allowed as defined by permission settings (UW, UR, SW, SR).
9	EXT		External Allowed ID.
		0	Requests with Privilege ID >= 6 are not allowed to region M, regardless of permission settings (UW, UR, SW, SR).
		1	Requests with Privilege ID >= 6 are permitted, if access type is allowed as defined by permission settings (UW, UR, SW, SR).
8	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
7-6	Reserved	1	Reserved. Always write 1 to this bit.
5	SR		Supervisor read permission.
		0	Supervisor read accesses are not allowed from region M.
		1	Supervisor write accesses are allowed from region M addresses.
4	SW		Supervisor write permission.
		0	Supervisor write accesses are not allowed to region M.
		1	Supervisor write accesses are allowed to region N addresses.
3	SX		Supervisor execute permission.
		0	Supervisor execute accesses are not allowed from region M.
		1	Supervisor execute accesses are allowed from region M addresses.
2	UR		User read permission.
		0	User read accesses are not allowed from region M.
		1	User read accesses are allowed from region N addresses.
1	UW		User write permission.
		0	User write accesses are not allowed to region M.
		1	User write accesses are allowed to region M addresses.
0	UX		User execute permission.
		0	User execute accesses are not allowed from region M.
		1	User execute accesses are allowed from region M addresses.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.6 DMA Channel Registers

The following sets of registers pertain to the 64 DMA channels. The 64 DMA channels consist of a set of registers (with exception of DMAQNUM*n*) that each have 64 bits and the bit position of each register matches the DMA channel number. Each register is named with the format *reg_name* that corresponds to DMA channels 0 through 31 and *reg_name_High* that corresponds to DMA channels 32 through 64.

For example, the event register (ER) corresponds to DMA channel 0 through 31 and the event register high register (ERH) corresponds to DMA channel 32 through 63. The register is typically called the event register.

The DMA channel registers are accessible via read/writes to the global address range. They are also accessible via read/writes to the shadow address range. The read/write ability to the registers in the shadow region are controlled by the DMA region access registers (DRAE*m*/DRAEH*m*). The registers are described in Section 10.5.2.3.1 and the details for shadow region/global region usage is explained in *EDMA3 Channel Controller Regions*.

10.5.2.6.1 Event Registers (ER, ERH)

All external events are captured in the event register (ER/ERH). The events are latched even when the events are not enabled. If the event bit corresponding to the latched event is enabled (EER.E*n*/EERH.E*n* = 1), then the event is evaluated by the EDMA3CC logic for an associated transfer request submission to the transfer controllers. The event register bits are automatically cleared (ER.E*n*/ERH.E*n* = 0) once the corresponding events are prioritized and serviced. If ER.E*n*/ERH.E*n* are already set and another event is received on the same channel/event, then the corresponding event is latched in the event miss register (EMR.E*n*/EMRH.E*n*), provided that the event was enabled (EER.E*n*/EERH.E*n* = 1).

Event *n* can be cleared by the CPU writing a 1 to corresponding event bit in the event clear register (ECR/ECRH). The setting of an event is a higher priority relative to clear operations (via hardware or software). If set and clear conditions occur concurrently, the set condition wins. If the event was previously set, then EMR/EMRH would be set because an event is lost. If the event was previously clear, then the event remains set and is prioritized for submission to the event queues.

The *Debug List* table provides the type of synchronization events and the EDMA3CC channels associated to each of these external events.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

The ER is shown in Figure 10-28 and described in Table 10-29. The ERH is shown in Figure 10-29 and described in Table 10-30.

Figure 10-28. Event Register (ER)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	19	E18	E17	E16
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 10-29. Event Register (ER) Field Descriptions

Bit	Field	Value	Description
31-0	E*n*		Event 0-31. Events 0-31 are captured by the EDMA3CC and are latched into ER. The events are set (E*n* = 1) even when events are disabled (E*n* = 0 in the event enable register, EER).
		0	EDMA3CC event is not asserted.
		1	EDMA3CC event is asserted. Corresponding DMA event is prioritized versus other pending DMA/QDMA events for submission to the EDMA3TC.

Figure 10-29. Event Register High (ERH)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 10-30. Event Register High (ERH) Field Descriptions

Bit	Field	Value	Description
31-0	E*n*		Event 32-63. Events 32-63 are captured by the EDMA3CC and are latched into ERH. The events are set (E*n* = 1) even when events are disabled (E*n* = 0 in the event enable register high, EERH).
		0	EDMA3CC event is not asserted.
		1	EDMA3CC event is asserted. Corresponding DMA event is prioritized versus other pending DMA/QDMA events for submission to the EDMA3TC.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.6.2 Event Clear Registers (ECR, ECRH)

Once an event has been posted in the event registers (ER/ERH), the event is cleared in two ways. If the event is enabled in the event enable register (EER/EERH) and the EDMA3CC submits a transfer request for the event to the EDMA3TC, it clears the corresponding event bit in the event register. If the event is disabled in the event enable register (EER/EERH), the CPU can clear the event by way of the event clear registers (ECR/ECRH).

Writing a 1 to any of the bits clears the corresponding event; writing a 0 has no effect. Once an event bit is set in the event register, it remains set until EDMA3CC submits a transfer request for that event or the CPU clears the event by setting the corresponding bit in ECR/ECRH.

The ECR is shown in Figure 10-30 and described in Table 10-31. The ECRH is shown in Figure 10-31 and described in Table 10-32.

Figure 10-30. Event Clear Register (ECR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0

LEGEND: W = Write only; -*n* = value after reset

Table 10-31. Event Clear Register (ECR) Field Descriptions

Bit	Field	Value	Description
31-0	E*n*		Event clear for event 0-31. Any of the event bits in ECR is set to clear the event (E*n*) in the event register (ER). A write of 0 has no effect.
		0	No effect.
		1	EDMA3CC event is cleared in the event register (ER).

Figure 10-31. Event Clear Register High (ECRH)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0

LEGEND: W = Write only; -*n* = value after reset

Table 10-32. Event Clear Register High (ECRH) Field Descriptions

Bit	Field	Value	Description
31-0	E*n*		Event clear for event 32-63. Any of the event bits in ECRH are set to clear the event (E*n*) in the event register high (ERH). A write of 0 has no effect.
		0	No effect.
		1	EDMA3CC event is cleared in the event register high (ERH).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.6.3 Event Set Registers (ESR, ESRH)

The event set registers (ESR/ESRH) allow the CPU (EDMA3 programmers) to manually set events to initiate DMA transfer requests. CPU writes of 1 to any event set register (E*n*) bits set the corresponding bits in the registers. The set event is evaluated by the EDMA3CC logic for an associated transfer request submission to the transfer controllers. Writing a 0 has no effect.

The event set registers operate independent of the event registers (ER/ERH), and a write of 1 is always considered a valid event regardless of whether the event is enabled (the corresponding event bits are set or cleared in EER.E*n*/EERH.E*n*).

Once the event is set in the event set registers, it cannot be cleared by CPU writes, in other words, the event clear registers (ECR/ECRH) have no effect on the state of ESR/ESRH. The bits will only be cleared once the transfer request corresponding to the event has been submitted to the transfer controller. The setting of an event is a higher priority relative to clear operations (via hardware). If set and clear conditions occur concurrently, the set condition wins. If the event was previously set, then EMR/EMRH would be set because an event is lost. If the event was previously clear, then the event remains set and is prioritized for submission to the event queues.

Manually-triggered transfers via writes to ESR/ESRH allow the CPU to submit DMA requests in the system, these are relevant for memory-to-memory transfer scenarios. If the ESR.E*n*/ESRH.E*n* bit is already set and another CPU write of 1 is attempted to the same bit, then the corresponding event is latched in the event missed registers (EMR.E*n*/EMRH.E*n* = 1).

The ESR is shown in Figure 10-32 and described in Table 10-33. The ESRH is shown in Figure 10-33 and described in Table 10-34.

Figure 10-32. Event Set Register (ESR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 10-33. Event Set Register (ESR) Field Descriptions

Bit	Field	Value	Description
31-0	E*n*		Event set for event 0-31.
		0	No effect.
		1	Corresponding DMA event is prioritized versus other pending DMA/QDMA events for submission to the EDMA3TC.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 10-33. Event Set Register High (ESRH)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 10-34. Event Set Register High (ESRH) Field Descriptions

Bit	Field	Value	Description
31-0	E*n*		Event set for event 32-63.
		0	No effect.
		1	Corresponding DMA event is prioritized versus other pending DMA/QDMA events for submission to the EDMA3TC.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.6.4 Chained Event Registers (CER, CERH)

When the OPTIONS parameter for a PaRAM entry is programmed to returned a chained completion code (ITCCHEN = 1 and/or TCCHEN = 1), then the value dictated by the TCC[5:0] (also programmed in OPT) forces the corresponding event bit to be set in the chained event registers (CER/CERH). The set chained event is evaluated by the EDMA3CC logic for an associated transfer request submission to the transfer controllers. This results in a chained-triggered transfer.

The chained event registers do not have any enables. The generation of a chained event is essentially enabled by the PaRAM entry that has been configured for intermediate and/or final chaining on transfer completion. The E*n* bit is set (regardless of the state of EER.E*n*/EERH.E*n*) when a chained completion code is returned from one of the transfer controllers or is generated by the EDMA3CC via the early completion path. The bits in the chained event register are cleared when the corresponding events are prioritized and serviced.

If the E*n* bit is already set and another chaining completion code is return for the same event, then the corresponding event is latched in the event missed registers (EMR.E*n*/EMRH.E*n* = 1). The setting of an event is a higher priority relative to clear operations (via hardware). If set and clear conditions occur concurrently, the set condition wins. If the event was previously set, then EMR/EMRH would be set because an event is lost. If the event was previously clear, then the event remains set and is prioritized for submission to the event queues.

The CER is shown in Figure 10-34 and described in Table 10-35. The CERH is shown in Figure 10-35 and described in Table 10-36.

Figure 10-34. Chained Event Register (CER)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 10-35. Chained Event Register (CER) Field Descriptions

Bit	Field	Value	Description
31-0	E*n*		Chained event for event 0-31.
		0	No effect.
		1	Corresponding DMA event is prioritized versus other pending DMA/QDMA events for submission to the EDMA3TC.

Figure 10-35. Chained Event Register High (CERH)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 10-36. Chained Event Register High (CERH) Field Descriptions

Bit	Field	Value	Description
31-0	E*n*		Chained event set for event 32-63.
		0	No effect.
		1	Corresponding DMA event is prioritized versus other pending DMA/QDMA events for submission to the EDMA3TC.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.6.5 Event Enable Registers (EER, EERH)

The EDMA3CC provides the option of selectively enabling/disabling each event in the event registers (ER/ERH) by using the event enable registers (EER/EERH). If an event bit in EER/EERH is set (using the event enable set registers, EESR/EESRH), it will enable that corresponding event. Alternatively, if an event bit in EER/EERH is cleared (using the event enable clear registers, EECR/EECRH), it will disable the corresponding event.

The event registers latch all events that are captured by EDMA3CC, even if the events are disabled (although EDMA3CC does not process it). Enabling an event with a pending event already set in the event registers enables the EDMA3CC to process the already set event like any other new event. The EER/EERH settings do not have any effect on chained events (CER.E*n*/CERH.E*n* = 1) and manually set events (ESR.E*n*/ESRH.E*n* = 1).

The EER is shown in Figure 10-36 and described in Table 10-37. Th EERH is shown in Figure 10-37 and described in Table 10-38.

Figure 10-36. Event Enable Register (EER)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 10-37. Event Enable Register (EER) Field Descriptions

Bit	Field	Value	Description
31-0	E*n*		Event enable for events 0-31.
		0	Event is not enabled. An external event latched in the event register (ER) is not evaluated by the EDMA3CC.
		1	Event is enabled. An external event latched in the event register (ER) is evaluated by the EDMA3CC.

Figure 10-37. Event Enable Register High (EERH)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 10-38. Event Enable Register High (EERH) Field Descriptions

Bit	Field	Value	Description
31-0	E*n*		Event enable for events 32-63.
		0	Event is not enabled. An external event latched in the event register high (ERH) is not evaluated by the EDMA3CC.
		1	Event is enabled. An external event latched in the event register high (ERH) is evaluated by the EDMA3CC.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.6.6 Event Enable Clear Register (EECR, EECRH)

The event enable registers (EER/EERH) cannot be modified by directly writing to them. The intent is to ease the software burden for the case where multiple tasks are attempting to simultaneously modify these registers. The event enable clear registers (EECR/EECRH) are used to disable events. Writes of 1 to the bits in EECR/EECRH clear the corresponding event bits in EER/EERH; writes of 0 have no effect.

The EECR is shown in Figure 10-38 and described in Table 10-39. The EECRH is shown in Figure 10-39 and described in Table 10-40.

Figure 10-38. Event Enable Clear Register (EECR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0

LEGEND: W = Write only; -*n* = value after reset

Table 10-39. Event Enable Clear Register (EECR) Field Descriptions

Bit	Field	Value	Description
31-0	E*n*		Event enable clear for events 0-31.
		0	No effect.
		1	Event is disabled. Corresponding bit in the event enable register (EER) is cleared (E*n* = 0).

Figure 10-39. Event Enable Clear Register High (EECRH)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0

LEGEND: W = Write only; -*n* = value after reset

Table 10-40. Event Enable Clear Register High (EECRH) Field Descriptions

Bit	Field	Value	Description
31-0	E*n*		Event enable clear for events 32-63.
		0	No effect.
		1	Event is disabled. Corresponding bit in the event enable register high (EERH) is cleared (E*n* = 0).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.6.7 Event Enable Set Registers (EESR, EESRH)

The event enable registers (EER/EERH) cannot be modified by directly writing to them. The intent is to ease the software burden for the case where multiple tasks are attempting to simultaneously modify these registers. The event enable set registers (EESR/EESRH) are used to enable events. Writes of 1 to the bits in EESR/EESRH set the corresponding event bits in EER/EERH; writes of 0 have no effect.

The EESR is shown in Figure 10-40 and described in Table 10-41. The EESRH is shown in Figure 10-41 and described in Table 10-42.

Figure 10-40. Event Enable Set Register (EESR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0

LEGEND: W = Write only; -*n* = value after reset

Table 10-41. Event Enable Set Register (EESR) Field Descriptions

Bit	Field	Value	Description
31-0	E*n*		Event enable set for events 0-31.
		0	No effect.
		1	Event is enabled. Corresponding bit in the event enable register (EER) is set (E*n* = 1).

Figure 10-41. Event Enable Set Register High (EESRH)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 10-42. Event Enable Set Register High (EESRH) Field Descriptions

Bit	Field	Value	Description
31-0	E*n*		Event enable set for events 32-63.
		0	No effect.
		1	Event is enabled. Corresponding bit in the event enable register high (EERH) is set (E*n* = 1).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.6.8 Secondary Event Registers (SER, SERH)

The secondary event registers (SER/SERH) provide information on the state of a DMA channel or event (0 through 63). If the EDMA3CC receives a TR synchronization due to a manual-trigger, event-trigger, or chained-trigger source (ESR.E*n*/ESRH.E*n* = 1, ER.E*n*/ERH.E*n* = 1, or CER.E*n*/CERH.E*n* = 1), which results in the setting of a corresponding event bit in SER/SERH (SER.E*n*/SERH.E*n* = 1), it implies that the corresponding DMA event is in the queue.

Once a bit corresponding to an event is set in SER/SERH, the EDMA3CC does not prioritize additional events on the same DMA channel. Depending on the condition that lead to the setting of the SER bits, either the EDMA3CC hardware or the software (using SECR/SECRH) needs to clear the SER/SERH bits for the EDMA3CC to evaluate subsequent events (subsequent transfers) on the same channel. For additional conditions that can cause the secondary event registers to be set, see *EDMA Overview*.

The SER is shown in Figure 10-42 and described in Table 10-43. The SERH is shown in Figure 10-43 and described in Table 10-44.

Figure 10-42. Secondary Event Register (SER)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 10-43. Secondary Event Register (SER) Field Descriptions

Bit	Field	Value	Description
31-0	E*n*		Secondary event register. The secondary event register is used along with the event register (ER) to provide information on the state of an event.
		0	Event is not currently stored in the event queue.
		1	Event is currently stored in the event queue. Event arbiter will not prioritize additional events.

Figure 10-43. Secondary Event Register High (SERH)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 10-44. Secondary Event Register High (SERH) Field Descriptions

Bit	Field	Value	Description
31-0	E*n*		Secondary event register. The secondary event register is used along with the event register high (ERH) to provide information on the state of an event.
		0	Event is not currently stored in the event queue.
		1	Event is currently stored in the event queue. Event submission/prioritization logic will not prioritize additional events.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.6.9 Secondary Event Clear Registers (SECR, SECRH)

The secondary event clear registers (SECR/SECRH) clear the status of the secondary event registers (SER/SERH). CPU writes of 1 clear the corresponding set bits in SER/SERH. Writes of 0 have no effect.

The SECR is shown in Figure 10-44 and described in Table 10-45. The SECRH is shown in Figure 10-45 and described in Table 10-46.

Figure 10-44. Secondary Event Clear Register (SECR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0

LEGEND: W = Write only; -*n* = value after reset

Table 10-45. Secondary Event Clear Register (SECR) Field Descriptions

Bit	Field	Value	Description
31-0	E*n*		Secondary event clear register.
		0	No effect.
		1	Corresponding bit in the secondary event register (SER) is cleared (E*n* = 0).

Figure 10-45. Secondary Event Clear Register High (SECRH)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0

LEGEND: W = Write only; -*n* = value after reset

Table 10-46. Secondary Event Clear Register High (SECRH) Field Descriptions

Bit	Field	Value	Description
31-0	E*n*		Secondary event clear register.
		0	No effect.
		1	Corresponding bit in the secondary event registers high (SERH) is cleared (E*n* = 0).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.7 Interrupt Registers

All DMA/QDMA channels can be set to assert an EDMA3CC completion interrupt to the CPU on transfer completion, by appropriately configuring the PaRAM entry associated with the channels. The following set of registers is used for the transfer completion interrupt reporting/generating by the EDMA3CC. For more details on EDMA3CC completion interrupt generation, see *EDMA3 Interrupts*.

10.5.2.7.1 Interrupt Enable Registers (IER, IERH)

Interrupt enable registers (IER/IERH) are used to enable/disable the transfer completion interrupt generation by the EDMA3CC for all DMA/QDMA channels. The IER/IERH cannot be written to directly. To set any interrupt bit in IER/IERH, a 1 must be written to the corresponding interrupt bit in the interrupt enable set registers (IESR/IESRH). Similarly, to clear any interrupt bit in IER/IERH, a 1 must be written to the corresponding interrupt bit in the interrupt enable clear registers (IECR/IECRH).

The IER is shown in Figure 10-46 and described in Table 10-47. The IERH is shown in Figure 10-47 and described in Table 10-48.

Figure 10-46. Interrupt Enable Register (IER)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I31	I30	I29	I28	I27I	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 10-47. Interrupt Enable Register (IER) Field Descriptions

Bit	Field	Value	Description
31-0	E*n*		Interrupt enable for channels 0-31.
		0	Interrupt is not enabled.
		1	Interrupt is enabled.

Figure 10-47. Interrupt Enable Register High (IERH)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 10-48. Interrupt Enable Register High (IERH) Field Descriptions

Bit	Field	Value	Description
31-0	E*n*		Interrupt enable for channels 32-63.
		0	Interrupt is not enabled.
		1	Interrupt is enabled.

10.5.2.7.2 Interrupt Enable Clear Register (IECR, IECRH)

The interrupt enable clear registers (IECR/IECRH) are used to clear interrupts. Writes of 1 to the bits in IECR/IECRH clear the corresponding interrupt bits in the interrupt enable registers (IER/IERH); writes of 0 have no effect.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

The IECR is shown in Figure 10-48 and described in Table 10-49. The IECRH is shown in Figure 10-49 and described in Table 10-50.

Figure 10-48. Interrupt Enable Clear Register (IECR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	16
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0

LEGEND: W = Write only; -*n* = value after reset

Table 10-49. Interrupt Enable Clear Register (IECR) Field Descriptions

Bit	Field	Value	Description
31-0	E*n*		Interrupt enable clear for channels 0-31.
		0	No effect.
		1	Corresponding bit in the interrupt enable register (IER) is cleared (I*n* = 0).

Figure 10-49. Interrupt Enable Clear Register High (IECRH)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0

LEGEND: W = Write only; -*n* = value after reset

Table 10-50. Interrupt Enable Clear Register High (IECRH) Field Descriptions

Bit	Field	Value	Description
31-0	E*n*		Interrupt enable clear for channels 32-63.
		0	No effect.
		1	Corresponding bit in the interrupt enable register high (IERH) is cleared (I*n* = 0).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.7.3 Interrupt Enable Set Registers (IESR, IESRH)

The interrupt enable set registers (IESR/IESRH) are used to enable interrupts. Writes of 1 to the bits in IESR/IESRH set the corresponding interrupt bits in the interrupt enable registers (IER/IERH); writes of 0 have no effect.

The IESR is shown in Figure 10-50 and described in Table 10-51. The IESRH is shown in Figure 10-51 and described in Table 10-52.

Figure 10-50. Interrupt Enable Set Register (IESR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	II8	I17	I16
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0

LEGEND: W = Write only; -*n* = value after reset

Table 10-51. Interrupt Enable Set Register (IESR) Field Descriptions

Bit	Field	Value	Description
31-0	E*n*		Interrupt enable set for channels 0-31.
		0	No effect.
		1	Corresponding bit in the interrupt enable register (IER) is set (I*n* = 1).

Figure 10-51. Interrupt Enable Set Register High (IESRH)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0

LEGEND: W = Write only; -*n* = value after reset

Table 10-52. Interrupt Enable Set Register High (IESRH) Field Descriptions

Bit	Field	Value	Description
31-0	E*n*		Interrupt enable clear for channels 32-63.
		0	No effect.
		1	Corresponding bit in the interrupt enable register high (IERH) is set (I*n* = 1).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.7.4 Interrupt Pending Register (IPR, IPRH)

If the TCINTEN and/or ITCINTEN bit in the channel option parameter (OPT) is set in the PaRAM entry associated with the channel (DMA or QDMA), then the EDMA3TC (for normal completion) or the EDMA3CC (for early completion) returns a completion code on transfer or intermediate transfer completion. The value of the returned completion code is equal to the TCC bit in OPT for the PaRAM entry associated with the channel.

When an interrupt transfer completion code with TCC = *n* is detected by the EDMA3CC, then the corresponding bit is set in the interrupt pending register (IPR.I*n*, if *n* = 0 to 31; IPRH.I*n*, if *n* = 32 to 63). Note that once a bit is set in the interrupt pending registers, it remains set; it is your responsibility to clear these bits. The bits set in IPR/IPRH are cleared by writing a 1 to the corresponding bits in the interrupt clear registers (ICR/ICRH).

The IPR is shown in Figure 10-52 and described in Table 10-53. The IPRH is shown in Figure 10-53 and described in Table 10-54.

Figure 10-52. Interrupt Pending Register (IPR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 10-53. Interrupt Pending Register (IPR) Field Descriptions

Bit	Field	Value	Description
31-0	I*n*		Interrupt pending for TCC = 0-31.
		0	Interrupt transfer completion code is not detected or was cleared.
		1	Interrupt transfer completion code is detected (I*n* = 1, *n* = EDMA3TC[5:0]).

Figure 10-53. Interrupt Pending Register High (IPRH)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 10-54. Interrupt Pending Register High (IPRH) Field Descriptions

Bit	Field	Value	Description
31-0	I*n*		Interrupt pending for TCC = 32-63.
		0	Interrupt transfer completion code is not detected or was cleared.
		1	Interrupt transfer completion code is detected (I*n* = 1, *n* = EDMA3TC[5:0]).

10.5.2.7.5 Interrupt Clear Registers (ICR, ICRH)

The bits in the interrupt pending registers (IPR/IPRH) are cleared by writing a 1 to the corresponding bits in the interrupt clear registers(ICR/ICRH). Writes of 0 have no effect. All set bits in IPR/IPRH must be cleared to allow EDMA3CC to assert additional transfer completion interrupts.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

The ICR is shown in Figure 10-54 and described in Table 10-55. The ICRH is shown in Figure 10-55 and described in Table 10-56.

Figure 10-54. Interrupt Clear Register (ICR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0

LEGEND: W = Write only; -*n* = value after reset

Table 10-55. Interrupt Clear Register (ICR) Field Descriptions

Bit	Field	Value	Description
31-0	I*n*		Interrupt clear register for TCC = 0-31.
		0	No effect.
		1	Corresponding bit in the interrupt pending register (IPR) is cleared (I*n* = 0).

Figure 10-55. Interrupt Clear Register High (ICRH)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0

LEGEND: W = Write only; -*n* = value after reset

Table 10-56. Interrupt Clear Register High (ICRH) Field Descriptions

Bit	Field	Value	Description
31-0	I*n*		Interrupt clear register for TCC = 32-63.
		0	No effect.
		1	Corresponding bit in the interrupt pending register high (IPRH) is cleared (I*n* = 0).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.7.6 Interrupt Evaluate Register (IEVAL)

The interrupt evaluate register (IEVAL) is the only register that physically exists in both the global region and the shadow regions. In other words, the read/write accessibility for the shadow region IEVAL is not affected by the DMA/QDMA region access registers (DRAE*m*/DRAEH*m*, QRAE*n*/QRAEH*n*). IEVAL is needed for robust ISR operations to ensure that interrupts are not missed by the CPU.

The IEVAL is shown in Figure 10-56 and described in Table 10-57.

Figure 10-56. Interrupt Evaluate Register (IEVAL)

31				16
Reserved				
R-0				

15	2	1	0
Reserved		Rsvd	EVAL
R-0		W-0	W-0

LEGEND: R = Read only; W = Write only; -*n* = value after reset

Table 10-57. Interrupt Evaluate Register (IEVAL) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
1	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
0	EVAL		Interrupt evaluate.
		0	No effect.
		1	Causes EDMA3CC completion interrupt to be pulsed, if any enabled (IER*n*/IERH*n* = 1) interrupts are still pending (IPR*n*/IPRH*n* = 1).
			The EDMA3CC completion interrupt that is pulsed depends on which IEVAL is being exercised. For example, writing to the EVAL bit in IEVAL pulses the global completion interrupt, but writing to the EVAL bit in IEVAL0 pulses the region 0 completion interrupt.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.8 QDMA Registers

The following sets of registers control the QDMA channels in the EDMA3CC. The QDMA channels (with the exception of the QDMA queue number register) consist of a set of registers, each of which have a bit location. Each bit position corresponds to a QDMA channel number. The QDMA channel registers are accessible via read/writes to the global address range. They are also accessible via read/writes to the shadow address range. The read/write accessibility in the shadow region address region is controlled by the QDMA region access registers (QRAE*n*/QRAEH*n*). *EDMA3 Channel Controller Regions* details shadow region/global region usage.

10.5.2.8.1 QDMA Event Register (QER)

The QDMA event register (QER) channel *n* bit is set (E*n* = 1) when the CPU or any EDMA3 programmer (including EDMA3) performs a write to the trigger word (using the QDMA channel mapping register (QCHMAP*n*)) in the PaRAM entry associated with QDMA channel *n* (which is also programmed using QCHMAP*n*). The E*n* bit is also set when the EDMA3CC performs a link update on a PaRAM address that matches the QCHMAP*n* settings. The QDMA event is latched only if the QDMA event enable register (QEER) channel *n* bit is also enabled (QEER.E*n* = 1). Once a bit is set in QER, then the corresponding QDMA event (auto-trigger) is evaluated by the EDMA3CC logic for an associated transfer request submission to the transfer controllers. For additional conditions that can lead to the setting of QER bits, see *EDMA Overview*.

The setting of an event is a higher priority relative to clear operations (via hardware). If set and clear conditions occur concurrently, the set condition wins. If the event was previously set, then the QDMA event missed register (QEMR) would be set because an event is lost. If the event was previously clear, then the event remains set and is prioritized for submission to the event queues.

The set bits in QER are only cleared when the transfer request associated with the corresponding channels has been processed by the EDMA3CC and submitted to the transfer controller. If the E*n* bit is already set and a QDMA event for the same QDMA channel occurs prior to the original being cleared, then the second missed event is latched in QEMR (E*n* = 1).

The QER is shown in Figure 10-57 and described in Table 10-58.

Figure 10-57. QDMA Event Register (QER)

31								16
Reserved								
R-0								

15	7	6	5	4	3	2	1	0
Reserved	E7	E6	E5	E4	E3	E2	E1	E0
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 10-58. QDMA Event Register (QER) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
7-0	E*n*		QDMA event for channels 0-7.
		0	No effect.
		1	Corresponding QDMA event is prioritized versus other pending DMA/QDMA events for submission to the EDMA3TC.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.8.2 QDMA Event Enable Register (QEER)

The EDMA3CC provides the option of selectively enabling/disabling each channel in the QDMA event register (QER) by using the QDMA event enable register (QEER). If any of the event bits in QEER is set (using the QDMA event enable set register, QEESR), it will enable that corresponding event. Alternatively, if any event bit in QEER is cleared (using the QDMA event enable clear register, QEECR), it will disable the corresponding QDMA channel. The QDMA event register will not latch any event for a QDMA channel, if it is not enabled via QEER.

The QEER is shown in Figure 10-58 and described in Table 10-59.

Figure 10-58. QDMA Event Enable Register (QEER)

31							16
Reserved							
R-0							

15	7	6	5	4	3	2	1	0
Reserved	E7	E6	E5	E4	E3	E2	E1	E0
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 10-59. QDMA Event Enable Register (QEER) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
7-0	E*n*		QDMA event enable for channels 0-7.
		0	QDMA channel *n* is not enabled. QDMA event will not be recognized and will not latch in the QDMA event register (QER).
		1	QDMA channel *n* is enabled. QDMA events will be recognized and will get latched in the QDMA event register (QER).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.8.3 QDMA Event Enable Clear Register (QEECR)

The QDMA event enable register (QEER) cannot be modified by directly writing to the register, to ease the software burden when multiple tasks are attempting to simultaneously modify these registers. The QDMA event enable clear register (QEECR) is used to disable events. Writes of 1 to the bits in QEECR clear the corresponding QDMA channel bits in QEER; writes of 0 have no effect.

The QEECR is shown in Figure 10-59 and described in Table 10-60.

Figure 10-59. QDMA Event Enable Clear Register (QEECR)

31							16
Reserved							
R-0							

15	7	6	5	4	3	2	1	0
Reserved	E7	E6	E5	E4	E3	E2	E1	E0
R-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0

LEGEND: R = Read only; W = Write only; -*n* = value after reset

Table 10-60. QDMA Event Enable Clear Register (QEECR) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
7-0	E*n*		QDMA event enable clear for channels 0-7.
		0	No effect.
		1	QDMA event is disabled. Corresponding bit in the QDMA event enable register (QEER) is cleared (E*n* = 0).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.8.4 QDMA Event Enable Set Register (QEESR)

The QDMA event enable register (QEER) cannot be modified by directly writing to the register, to ease the software burden when multiple tasks are attempting to simultaneously modify these registers. The QDMA event enable set register (QEESR) is used to enable events. Writes of 1 to the bits in QEESR set the corresponding QDMA channel bits in QEER; writes of 0 have no effect.

The QEESR is shown in Figure 10-60 and described in Table 10-61.

Figure 10-60. QDMA Event Enable Set Register (QEESR)

31							16
Reserved							
R-0							

15	7	6	5	4	3	2	1	0
Reserved	E7	E6	E5	E4	E3	E2	E1	E0
R-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0

LEGEND: R = Read only; W = Write only; -*n* = value after reset

Table 10-61. QDMA Event Enable Set Register (QEESR) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
7-0	E*n*		QDMA event enable set for channels 0-7.
		0	No effect.
		1	QDMA event is enabled. Corresponding bit in the QDMA event enable register (QEER) is set (E*n* = 1).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.8.5 QDMA Secondary Event Register (QSER)

The QDMA secondary event register (QSER) provides information on the state of a QDMA event. If at any time a bit corresponding to a QDMA channel is set in QSER, that implies that the corresponding QDMA event is in the queue. Once a bit corresponding to a QDMA channel is set in QSER, the EDMA3CC does not prioritize additional events on the same QDMA channel. Depending on the condition that lead to the setting of the QSER bits, either the EDMA3CC hardware or the software (using QSECR) needs to clear the QSER bits for the EDMA3CC to evaluate subsequent QDMA events on the channel. Based on whether the associated TR request is valid, or it is a null or dummy TR, the implications on the state of QSER and the required user actions to submit another QDMA transfer might be different. For additional conditions that can cause the secondary event registers (QSER\SER) to be set, see *EDMA Overview*.

The QSER is shown in Figure 10-61 and described in Table 10-62.

Figure 10-61. QDMA Secondary Event Register (QSER)

31								16
Reserved								
R-0								

15	7	6	5	4	3	2	1	0
Reserved	E7	E6	E5	E4	E3	E2	E1	E0
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 10-62. QDMA Secondary Event Register (QSER) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
7-0	E*n*		QDMA secondary event register for channels 0-7.
		0	QDMA event is not currently stored in the event queue.
		1	QDMA event is currently stored in the event queue. EDMA3CC will not prioritize additional events.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.2.8.6 QDMA Secondary Event Clear Register (QSECR)

The QDMA secondary event clear register (QSECR) clears the status of the QDMA secondary event register (QSER) and the QDMA event register (QER). CPU writes of 1 clear the corresponding set bits in QSER and QER. Writes of 0 have no effect. Note that this differs from the secondary event clear register (SECR) operation, which only clears the secondary event register (SER) bits and does not affect the event registers.

The QSECR is shown in Figure 10-62 and described in Table 10-63.

Figure 10-62. QDMA Secondary Event Clear Register (QSECR)

31								16
Reserved								
R-0								

15	7	6	5	4	3	2	1	0
Reserved	E7	E6	E5	E4	E3	E2	E1	E0
R-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0

LEGEND: R = Read only; W = Write only; -*n* = value after reset

Table 10-63. QDMA Secondary Event Clear Register (QSECR) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-0	E*n*		QDMA secondary event clear register for channels 0-7.
		0	No effect.
		1	Corresponding bit in the QDMA secondary event register (QSER) and the QDMA event register (QER) is cleared (E*n* = 0).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.3 EDMA3 Transfer Controller Control Registers

Table 10-64 lists the memory-mapped registers for the EDMA3 transfer controller (EDMA3TC). See the device-specific data manual for the memory address of these registers.

Table 10-64. EDMA3 Transfer Controller Registers

Offset	Acronym	Register Description	Section
00h	PID	Peripheral Identification Register	Section 10.5.3.1
04h	TCCFG	EDMA3TC Configuration Register	Section 10.5.3.2
0100h	TCSTAT	EDMA3TC Channel Status Register	Section 10.5.3.3
0120h	ERRSTAT	Error Register	Section 10.5.3.4.1
0124h	ERREN	Error Enable Register	Section 10.5.3.4.2
0128h	ERRCLR	Error Clear Register	Section 10.5.3.4.3
012Ch	ERRDET	Error Details Register	Section 10.5.3.4.4
0130h	ERRCMD	Error Interrupt Command Register	Section 10.5.3.4.5
0140h	RDRATE	Read Rate Register	Section 10.5.3.5
0240h	SAOPT	Source Active Options Register	Section 10.5.3.6.1
0244h	SASRC	Source Active Source Address Register	Section 10.5.3.6.2
0248h	SACNT	Source Active Count Register	Section 10.5.3.6.3
024Ch	SADST	Source Active Destination Address Register	Section 10.5.3.6.4
0250h	SABIDX	Source Active Source B-Index Register	Section 10.5.3.6.5
0254h	SAMPPRXY	Source Active Memory Protection Proxy Register	Section 10.5.3.6.6
0258h	SACNTRLD	Source Active Count Reload Register	Section 10.5.3.6.7
025Ch	SASRCBREF	Source Active Source Address B-Reference Register	Section 10.5.3.6.8
0260h	SADSTBREF	Source Active Destination Address B-Reference Register	Section 10.5.3.6.9
0280h	DFCNTRLD	Destination FIFO Set Count Reload	Section 10.5.3.6.16
0284h	DFSRCBREF	Destination FIFO Set Destination Address B Reference Register	Section 10.5.3.6.17
0288h	DFDSTBREF	Destination FIFO Set Destination Address B Reference Register	Section 10.5.3.6.18
0300h	DFOPT0	Destination FIFO Options Register 0	Section 10.5.3.6.10
0304h	DFSRC0	Destination FIFO Source Address Register 0	Section 10.5.3.6.11
0308h	DFCNT0	Destination FIFO Count Register 0	Section 10.5.3.6.12
030Ch	DFDST0	Destination FIFO Destination Address Register 0	Section 10.5.3.6.13
0310h	DFBIDX0	Destination FIFO BIDX Register 0	Section 10.5.3.6.14
0314h	DFMPPRXY0	Destination FIFO Memory Protection Proxy Register 0	Section 10.5.3.6.15
0340h	DFOPT1	Destination FIFO Options Register 1	Section 10.5.3.6.10
0344h	DFSRC1	Destination FIFO Source Address Register 1	Section 10.5.3.6.11
0348h	DFCNT1	Destination FIFO Count Register 1	Section 10.5.3.6.12
034Ch	DFDST1	Destination FIFO Destination Address Register 1	Section 10.5.3.6.13
0350h	DFBIDX1	Destination FIFO BIDX Register 1	Section 10.5.3.6.14
0354h	DFMPPRXY1	Destination FIFO Memory Protection Proxy Register 1	Section 10.5.3.6.15
0380h	DFOPT2	Destination FIFO Options Register 2	Section 10.5.3.6.10
0384h	DFSRC2	Destination FIFO Source Address Register 2	Section 10.5.3.6.11
0388h	DFCNT2	Destination FIFO Count Register 2	Section 10.5.3.6.12
038Ch	DFDST2	Destination FIFO Destination Address Register 2	Section 10.5.3.6.13
0390h	DFBIDX2	Destination FIFO BIDX Register 2	Section 10.5.3.6.14
0394h	DFMPPRXY2	Destination FIFO Memory Protection Proxy Register 2	Section 10.5.3.6.15
03C0h	DFOPT3	Destination FIFO Options Register 3	Section 10.5.3.6.10
03C4h	DFSRC3	Destination FIFO Source Address Register 3	Section 10.5.3.6.11
03C8h	DFCNT3	Destination FIFO Count Register 3	Section 10.5.3.6.12
03CCh	DFDST3	Destination FIFO Destination Address Register 3	Section 10.5.3.6.13

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 10-64. EDMA3 Transfer Controller Registers (continued)

Offset	Acronym	Register Description	Section
03D0h	DFBIDX3	Destination FIFO BIDX Register 3	Section 10.5.3.6.14
03D4h	DFMPPRXY3	Destination FIFO Memory Protection Proxy Register 3	Section 10.5.3.6.15

10.5.3.1 Peripheral Identification Register (PID)

The peripheral identification register (PID) is a constant register that uniquely identifies the EDMA3TC and specific revision of the EDMA3TC. The PID is shown in Figure 10-63 and described in Table 10-65.

Figure 10-63. Peripheral ID Register (PID)

31 – 16	15 – 0
PID	PID
R-0333h	R-4425h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 10-65. Peripheral ID Register (PID) Field Descriptions

Bit	Field	Value	Description
31-0	PID	0-FFFF FFFFh	Peripheral identifier.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.3.2 EDMA3TC Configuration Register (TCCFG)

The EDMA3TC configuration register (TCCFG) is shown in Figure 10-64 and described in Table 10-66.

Figure 10-64. EDMA3TC Configuration Register (TCCFG)

31–16	15–10	9–8	7–6	5–4	3	2–0
Reserved	Reserved	DREGDEPTH	Reserved	BUSWIDTH	Rsvd	FIFOSIZE
R-0	R-0	R-x	R-0	R-n	R-0	R-x

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset; -x = value is indeterminate after reset

Table 10-66. EDMA3TC Configuration Register (TCCFG) Field Descriptions

Bit	Field	Value	Description
31-10	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
9-8	DREGDEPTH	0-3h	Destination register FIFO depth parameterization.
		0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
		1h	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
		2h	4 entry (for TC0, TC1, TC2, and TC3)
		3h	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
7-6	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
5-4	BUSWIDTH	0-3h	Bus width parameterization.
		0-1h	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
		2h	128-bit
		3h	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
3	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
2-0	FIFOSIZE	0-7h	FIFO size
		0-1h	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
		2h	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
		3h	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
		4h	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
		5h	1024 byte FIFO (for TC0, TC1, TC2 and TC3)
		6h-7h	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.3.3 EDMA3TC Channel Status Register (TCSTAT)

The EDMA3TC channel status register (TCSTAT) is shown in Figure 10-65 and described in Table 10-67.

Figure 10-65. EDMA3TC Channel Status Register (TCSTAT)

31–16	15–14	13–12	11–9	8	7	6–4	3	2	1	0
Reserved	Reserved	DFSTRTPTR	Reserved	Reserved	Reserved	DSTACTV	Reserved	WSACTV	SRCACTV	PROGBUSY
R-0	R-0	R-0	R-0	R-1	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 10-67. EDMA3TC Channel Status Register (TCSTAT) Field Descriptions

Bit	Field	Value	Description
31-14	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
13-12	DFSTRTPTR	0-3h	Destination FIFO start pointer. Represents the offset to the head entry of the destination register FIFO, in units of entries.
11-9	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
8	Reserved	1	Reserved. Always read as 1.
7	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
6-4	DSTACTV	0-7h	Destination active state. Specifies the number of transfer requests (TRs) that are resident in the destination register FIFO at a given instant. This bit field can be primarily used for advanced debugging. Legal values are constrained by the destination register FIFO depth parameterization (DSTREGDEPTH) parameter.
		0	Destination FIFO is empty.
		1h	Destination FIFO contains 1 TR.
		2h	Destination FIFO contains 2 TRs.
		3h	Destination FIFO contains 3 TRs.
		4h	Destination FIFO contains 4 TRs. (Full if DSTREGDEPTH==4).
			If the destination register FIFO is empty, then any TR written to Prog Set immediately transitions to the destination register FIFO. If the destination register FIFO is not empty and not full, then any TR written to Prog Set immediately transitions to the destination register FIFO set if the source active state (SRCACTV) bit is set to idle.
			If the destination register FIFO is full, then TRs cannot transition to the destination register FIFO. The destination register FIFO becomes not full when the TR at the head of the destination register FIFO is completed.
		5h-7h	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
3	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
2	WSACTV		Write status active
		0	Write status is not pending. Write status has been received for all previously issued write commands.
		1	Write status is pending. Write status has not been received for all previously issued write commands.
1	SRCACTV		Source active state
		0	Source controller is idle. Source active register set contains a previously processed transfer request.
		1	Source controller is busy servicing a transfer request.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 10-67. EDMA3TC Channel Status Register (TCSTAT) Field Descriptions (continued)

Bit	Field	Value	Description
0	PROGBUSY		Program register set busy
		0	Program set idle and is available for programming by the EDMA3CC.
		1	Program set busy

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.3.4 Error Registers

10.5.3.4.1 Error Register (ERRSTAT)

The error status register (ERRSTAT) is shown in Figure 10-66 and described in Table 10-68.

Figure 10-66. Error Register (ERRSTAT)

31–16	15–4	3	2	1	0
Reserved	Reserved	MMRAERR	TRERR	Reserved	BUSERR
R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 10-68. Error Register (ERRSTAT) Field Descriptions

Bit	Field	Value	Description
31-4	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
3	MMRAERR		MMR address error.
		0	Condition is not detected.
		1	User attempted to read or write to an invalid address in configuration memory map.
2	TRERR		Transfer request (TR) error event.
		0	Condition is not detected.
		1	TR detected that violates constant addressing mode transfer (SAM or DAM is set) alignment rules or has ACNT or BCNT == 0.
1	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
0	BUSERR		Bus error event.
		0	Condition is not detected.
		1	EDMA3TC has detected an error at source or destination address. Error information can be read from the error details register (ERRDET).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.3.4.2 Error Enable Register (ERREN)

The error enable register (ERREN) is shown in Figure 10-67 and described in Table 10-69.

When any of the enable bits are set, a bit set in the corresponding ERRSTAT causes an assertion of the EDMA3TC interrupt.

Figure 10-67. Error Enable Register (ERREN)

31–16	15–4	3	2	1	0
Reserved	Reserved	MMRAERR	TRERR	Reserved	BUSERR
R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 10-69. Error Enable Register (ERREN) Field Descriptions

Bit	Field	Value	Description
31-4	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
3	MMRAERR		Interrupt enable for MMR address error (MMRAERR).
		0	MMRAERR is disabled.
		1	MMRAERR is enabled and contributes to the state of EDMA3TC error interrupt generation
2	TRERR		Interrupt enable for transfer request error (TRERR).
		0	TRERR is disabled.
		1	TRERR is enabled and contributes to the state of EDMA3TC error interrupt generation.
1	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
0	BUSERR		Interrupt enable for bus error (BUSERR).
		0	BUSERR is disabled.
		1	BUSERR is enabled and contributes to the state of EDMA3TC error interrupt generation.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.3.4.3 Error Clear Register (ERRCLR)

The error clear register (ERRCLR) is shown in Figure 10-68 and described in Table 10-70.

Figure 10-68. Error Clear Register (ERRCLR)

31–16	15–4	3	2	1	0
Reserved	Reserved	MMRAERR	TRERR	Reserved	BUSERR
R-0	R-0	W-0	W-0	R-0	W-0

LEGEND: R = Read only; W = Write only; -*n* = value after reset

Table 10-70. Error Clear Register (ERRCLR) Field Descriptions

Bit	Field	Value	Description
31-4	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
3	MMRAERR		Interrupt enable clear for the MMRAERR bit in the error status register (ERRSTAT).
		0	No effect.
		1	Clears the MMRAERR bit in ERRSTAT but does not clear the error details register (ERRDET).
2	TRERR		Interrupt enable clear for the TRERR bit in the error status register (ERRSTAT).
		0	No effect.
		1	Clears the TRERR bit in ERRSTAT but does not clear the error details register (ERRDET).
1	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
0	BUSERR		Interrupt clear for the BUSERR bit in the error status register (ERRSTAT).
		0	No effect.
		1	Clears the BUSERR bit in ERRSTAT and clears the error details register (ERRDET).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.3.4.4 Error Details Register (ERRDET)

The error details register (ERRDET) is shown in Figure 10-69 and described in Table 10-71.

Figure 10-69. Error Details Register (ERRDET)

31	18	17	16
Reserved		TCCHEN	TCINTEN
R-0		R-0	R-0

15	14	13	8	7	4	3	0
Reserved		TCC		Reserved		STAT	
R-0		R-0		R-0		R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 10-71. Error Details Register (ERRDET) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
17	TCCHEN	0-1	Transfer completion chaining enable. Contains the TCCHEN value in the channel options parameter (OPT) programmed by the channel controller for the read or write transaction that resulted in an error.
16	TCINTEN	0-1	Transfer completion interrupt enable. Contains the TCINTEN value in the channel options parameter (OPT) programmed by the channel controller for the read or write transaction that resulted in an error.
15-14	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
13 - 8	TCC	0-3Fh	Transfer complete code. Contains the TCC value in the channel options parameter (OPT) programmed by the channel controller for the read or write transaction that resulted in an error.
7-4	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
3-0	STAT	0-Fh	Transaction status. Stores the nonzero status/error code that was detected on the read status or write status bus. If read status and write status are returned on the same cycle, then the EDMA3TC chooses nonzero version. If both are nonzero, then the write status is treated as higher priority.
		0	No error.
		1h-7h	Read error.
		8h-Fh	Write error.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.3.4.5 Error Interrupt Command Register (ERRCMD)

The error command register (ERRCMD) is shown in Figure 10-70 and described in Table 10-72.

Figure 10-70. Error Interrupt Command Register (ERRCMD)

31	16
Reserved	
R-0	

15	2	1	0
Reserved		Rsvd	EVAL
R-0		W-0	W-0

LEGEND: R = Read only; W = Write only; -*n* = value after reset

Table 10-72. Error Interrupt Command Register (ERRCMD) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
1	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
0	EVAL		Error evaluate
		0	No effect
		1	EDMA3TC error line is pulsed if any of the error status register (ERRSTAT) bits are set.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.3.5 Read Rate Register (RDRATE)

The EDMA3 transfer controller issues read commands at a rate controlled by the read rate register (RDRATE). The RDRATE defines the number of idle cycles that the read controller must wait before issuing subsequent commands. This applies both to commands within a transfer request packet (TRP) and for commands that are issued for different transfer requests (TRs). For instance, if RDRATE is set to 4 cycles between reads, there are 3 inactive cycles between reads.

RDRATE allows flexibility in transfer controller access requests to an endpoint. For an application, RDRATE can be manipulated to slow down the access rate, so that the endpoint may service requests from other masters during the inactive EDMA3TC cycles.

The RDRATE is shown in Figure 10-71 and described in Table 10-73.

NOTE: It is expected that the RDRATE value for a transfer controller is static, as it is decided based on the application requirement. It is not recommended to change this setting on the fly.

Figure 10-71. Read Rate Register (RDRATE)

31			16
Reserved			
R-0			

15	3	2	0
Reserved		RDRATE	
R-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 10-73. Read Rate Register (RDRATE) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
2-0	RDRATE	0-7h	Read rate. Controls the number of cycles between read commands. This is a global setting that applies to all TRs for this EDMA3TC.
		0	Reads issued as fast as possible.
		1h	4 cycles between reads.
		2h	8 cycles between reads.
		3h	16 cycles between reads.
		4h	32 cycles between reads.
		5h-7h	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.3.6 EDMA3TC Channel Registers

The EDMA3TC channel registers are split into three parts: the programming registers, the source active registers, and the destination FIFO register. This section describes the registers and their functions. The program register set is programmed by the channel controller, and is for internal use. The other two sets are read-only and provided to facilitate advanced debug capabilities. The number of destination FIFO register sets depends on the destination FIFO depth.

TC0 has a FIFO depth of 2, so there are two sets of destination FIFO registers associated with TC0. TC1, TC2, and TC3 have a destination FIFO depth of 4, so there are four sets of destination FIFO registers for each of these transfer controllers.

10.5.3.6.1 Source Active Options Register (SAOPT)

The source active options register (SAOPT) is shown in Figure 10-72 and described in Table 10-74.

Figure 10-72. Source Active Options Register (SAOPT)

31–23	22	21	20	19–18	17–16
Reserved	TCCHEN	Rsvd	TCINTEN	Reserved	TCC
R-0	R/W-0	R-0	R/W-0	R-0	R/W-0

15–12	11	10–8	7	6–4	3–2	1	0
TCC	Rsvd	FWID	Rsvd	PRI	Reserved	DAM	SAM
R/W-0	R-0	R/W-0	R-0	R/W-0	R-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 10-74. Source Active Options Register (SAOPT) Field Descriptions

Bit	Field	Value	Description
31-23	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
22	TCCHEN		Transfer complete chaining enable
		0	Transfer complete chaining is disabled.
		1	Transfer complete chaining is enabled.
21	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
20	TCINTEN		Transfer complete interrupt enable.
		0	Transfer complete interrupt is disabled.
		1	Transfer complete interrupt is enabled.
19-18	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
17-12	TCC	0-3Fh	Transfer complete code. This 6-bit code is used to set the relevant bit in CER or IPR of the EDMA3PCC module.
11	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
10-8	FWID	0-7h	FIFO width. Applies if either SAM or DAM is set to constant addressing mode.
		0	FIFO width is 8-bit.
		1h	FIFO width is 16-bit.
		2h	FIFO width is 32-bit.
		3h	FIFO width is 64-bit.
		4h	FIFO width is 128-bit.
		5h	FIFO width is 256-bit.
		6h-7h	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
7	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 10-74. Source Active Options Register (SAOPT) Field Descriptions (continued)

Bit	Field	Value	Description
6-4	PRI	0-7h	Transfer priority. Reflects the values programmed in the QUEPRI register in the EDMACC.
		0	Priority 0 - Highest priority
		1h-6h	Priority 1 to priority 6
		7h	Priority 7 - Lowest priority
3-2	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
1	DAM		Destination address mode within an array
		0	Increment (INCR) mode. Destination addressing within an array increments.
		1	Constant addressing (CONST) mode. Destination addressing within an array wraps around upon reaching FIFO width.
0	SAM		Source address mode within an array
		0	Increment (INCR) mode. Source addressing within an array increments.
		1	Constant addressing (CONST) mode. Source addressing within an array wraps around upon reaching FIFO width.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.3.6.2 Source Active Source Address Register (SASRC)

The source active source address register (SASRC) is shown in Figure 10-73 and described in Table 10-75.

Figure 10-73. Source Active Source Address Register (SASRC)

31	0
SADDR	
R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 10-75. Source Active Source Address Register (SASRC) Field Descriptions

Bit	Field	Value	Description
31-0	SADDR	0-FFFF FFFFh	Source address for program register set. EDMA3TC updates value according to source addressing mode (SAM bit in the source active options register, SAOPT) .

10.5.3.6.3 Source Active Count Register (SACNT)

The source active count register (SACNT) is shown in Figure 10-74 and described in Table 10-76.

Figure 10-74. Source Active Count Register (SACNT)

31	16
BCNT	
R-0	

15	0
ACNT	
R-0	

LEGEND: R = Read only; -*n* = value after reset

Table 10-76. Source Active Count Register (SACNT) Field Descriptions

Bit	Field	Value	Description
31-16	BCNT	0-FFFFh	B dimension count. Number of arrays to be transferred, where each array is ACNT in length. It is decremented after each read command appropriately. Represents the amount of data remaining to be read. It should be 0 when transfer request (TR) is complete.
15-0	ACNT	0-FFFFh	A dimension count. Number of bytes to be transferred in first dimension. It is decremented after each read command appropriately. Represents the amount of data remaining to be read. It should be 0 when transfer request (TR) is complete.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.3.6.4 Source Active Destination Address Register (SADST)

The source active destination address register (SADST) is shown in Figure 10-75 and described in Table 10-77.

Figure 10-75. Source Active Destination Address Register (SADST)

31	16
Reserved	
R-0	

15	0
Reserved	
R-0	

LEGEND: R = Read only; -*n* = value after reset

Table 10-77. Source Active Destination Address Register (SADST) Field Descriptions

Bit	Field	Value	Description
31-0	Reserved	0	Reserved. Always reads as 0.

10.5.3.6.5 Source Active Source B-Dimension Index Register (SABIDX)

The source active set B-dimension index register (SABIDX) is shown in Figure 10-76 and described in Table 10-78.

Figure 10-76. Source Active Source B-Dimension Index Register (SABIDX)

31	16
DBIDX	
R-0	

15	0
SBIDX	
R-0	

LEGEND: R = Read only; -*n* = value after reset

Table 10-78. Source Active Source B-Dimension Index Register (SABIDX) Field Descriptions

Bit	Field	Value	Description
31-16	DBIDX	0	B-Index offset between destination arrays. Represents the offset in bytes between the starting address of each destination. Always reads as 0.
15-0	SBIDX	0-FFFFh	B-Index offset between source arrays. Represents the offset in bytes between the starting address of each source array.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.3.6.6 Source Active Memory Protection Proxy Register (SAMPPRXY)

The source active memory protection proxy register (SAMPPRXY) is shown in Figure 10-77 and described in Table 10-79.

Figure 10-77. Source Active Memory Protection Proxy Register (SAMPPRXY)

31				16
Reserved				
R-0				

15	9	8	7	4	3	0
Reserved		PRIV	Reserved		PRIVID	
R-0		R-0	R-0		R-0	

LEGEND: R = Read only; -*n* = value after reset

Table 10-79. Source Active Memory Protection Proxy Register (SAMPPRXY) Field Descriptions

Bit	Field	Value	Description
31-9	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
8	PRIV		Privilege level. The privilege level used by the host to set up the parameter entry in the channel controller. This field is set up when the associated TR is submitted to the EDMA3TC. The privilege ID is used while issuing read and write command to the target endpoints so that the target endpoints can perform memory protection checks based on the PRIV of the host that set up the DMA transaction.
		0	User-level privilege.
		1	Supervisor-level privilege.
7-4	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
3-0	PRIVID	0-Fh	Privilege ID. This contains the privilege ID of the host that set up the parameter entry in the channel controller. This field is set up when the associated TR is submitted to the EDMA3TC. This PRIVID value is used while issuing read and write commands to the target endpoints so that the target endpoints can perform memory protection checks based on the PRIVID of the host that set up the DMA transaction.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.3.6.7 Source Active Count Reload Register (SACNTRLD)

The source active count reload register (SACNTRLD) is shown in Figure 10-78 and described in Table 10-80.

Figure 10-78. Source Active Count Reload Register (SACNTRLD)

31	16
Reserved	
R-0	

15	0
ACNTRLD	
R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 10-80. Source Active Count Reload Register (SACNTRLD) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
15-0	ACNTRLD	0-FFFFh	A-count reload value. Represents the originally programmed value of ACNT. The reload value is used to reinitialize ACNT after each array is serviced.

10.5.3.6.8 Source Active Source Address B-Reference Register (SASRCBREF)

The source active source address B-reference register (SASRCBREF) is shown in Figure 10-79 and described in Table 10-81.

Figure 10-79. Source Active Source Address B-Reference Register (SASRCBREF)

31	16
SADDRBREF	
R-0	

15	0
SADDRBREF	
R-0	

LEGEND: R = Read only; -*n* = value after reset

Table 10-81. Source Active Source Address B-Reference Register (SASRCBREF) Field Descriptions

Bit	Field	Value	Description
31-0	SADDRBREF	0-FFFF FFFFh	Source address B-reference. Represents the starting address for the array currently being read.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.3.6.9 Source Active Destination Address B-Reference Register (SADSTBREF)

The source active destination address B-reference register (SADSTBREF) is shown in Figure 10-80 and described in Table 10-82.

Figure 10-80. Source Active Destination Address B-Reference Register (SADSTBREF)

31	16
Reserved	
R-0	

15	0
Reserved	
R-0	

LEGEND: R = Read only; -*n* = value after reset

Table 10-82. Source Active Destination Address B-Reference Register (SADSTBREF) Field Descriptions

Bit	Field	Value	Description
31-0	Reserved	0	Reserved. Always reads as 0.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.3.6.10 Destination FIFO Options Register (DFOPTn)

The destination FIFO options register (DFOPT*n*) is shown in Figure 10-81 and described in Table 10-83.

NOTE: The value for *n* varies from 0 to DSTREGDEPTH for the given EDMA3TC.

Figure 10-81. Destination FIFO Options Register (DFOPT*n*)

31–23	22	21	20	19–18	17–16
Reserved	TCCHEN	Rsvd	TCINTEN	Reserved	TCC
R-0	R/W-0	R-0	R/W-0	R-0	R/W-0

15–12	11	10–8	7	6–4	3–2	1	0
TCC	Rsvd	FWID	Rsvd	PRI	Reserved	DAM	SAM
R/W-0	R-0	R/W-0	R-0	R/W-0	R-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 10-83. Destination FIFO Options Register (DFOPT*n*) Field Descriptions

Bit	Field	Value	Description
31-23	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
22	TCCHEN		Transfer complete chaining enable
		0	Transfer complete chaining is disabled
		1	Transfer complete chaining is enabled
21	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
20	TCINTEN		Transfer complete interrupt enable.
		0	Transfer complete interrupt is disabled.
		1	Transfer complete interrupt is enabled.
19-18	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
17-12	TCC	0-3Fh	Transfer complete code. This 6-bit code is used to set the relevant bit in CER or IPR of the EDMA3PCC module.
11	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
10-8	FWID	0-7h	FIFO width. Applies if either SAM or DAM is set to constant addressing mode.
		0	FIFO width is 8-bit.
		1h	FIFO width is 16-bit.
		2h	FIFO width is 32-bit.
		3h	FIFO width is 64-bit.
		4h	FIFO width is 128-bit.
		5h	FIFO width is 256-bit.
		6h-7h	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
7	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
6-4	PRI	0-7h	Transfer priority
		0	Priority 0 - Highest priority
		1h-6h	Priority 1 to priority 6
		7h	Priority 7 - Lowest priority
3-2	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 10-83. Destination FIFO Options Register (DFOPT*n*) Field Descriptions (continued)

Bit	Field	Value	Description
1	DAM		Destination address mode within an array
		0	Increment (INCR) mode. Destination addressing within an array increments.
		1	Constant addressing (CONST) mode. Destination addressing within an array wraps around upon reaching FIFO width.
0	SAM		Source address mode within an array
		0	Increment (INCR) mode. Source addressing within an array increments.
		1	Constant addressing (CONST) mode. Source addressing within an array wraps around upon reaching FIFO width.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.3.6.11 Destination FIFO Source Address Register (DFSRC*n*)

The destination FIFO source address register (DFSRC*n*) is shown in Figure 10-82 and described in Table 10-84.

NOTE: The value for *n* varies from 0 to DSTREGDEPTH for the given EDMA3TC.

Figure 10-82. Destination FIFO Source Address Register (DFSRC*n*)

31	16
Reserved	
R-0	

15	0
Reserved	
R-0	

LEGEND: R = Read only; -*n* = value after reset

Table 10-84. Destination FIFO Source Address Register (DFSRC*n*) Field Descriptions

Bit	Field	Value	Description
31-0	Reserved	0	Reserved. Always read as 0.

10.5.3.6.12 Destination FIFO Count Register (DFCNT*n*)

The destination FIFO count register (DFCNT*n*) is shown in Figure 10-83 and described in Table 10-85.

NOTE: The value for *n* varies from 0 to DSTREGDEPTH for the given EDMA3TC.

Figure 10-83. Destination FIFO Count Register (DFCNT*n*)

31	16
BCNT	
R-0	

15	0
ACNT	
R-0	

LEGEND: R = Read only; -*n* = value after reset

Table 10-85. Destination FIFO Count Register (DFCNT*n*) Field Descriptions

Bit	Field	Value	Description
31-16	BCNT	0-FFFFh	B-dimension count. Number of arrays to be transferred, where each array is ACNT in length. Count/count remaining for destination register set. Represents the amount of data remaining to be written.
15-0	ACNT	0-FFFFh	A-dimension count. Number of bytes to be transferred in first dimension count/count remaining for destination register set. Represents the amount of data remaining to be written.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.3.6.13 Destination FIFO Destination Address Register (DFDSTn)

The destination FIFO destination address register (DFDSTn) is shown in Figure 10-84 and described in Table 10-86.

NOTE: The value for *n* varies from 0 to DSTREGDEPTH for the given EDMA3TC.

Figure 10-84. Destination FIFO Destination Address Register (DFDSTn)

31	16
DADDR	
R-0	

15	0
DADDR	
R-0	

LEGEND: R = Read only; -*n* = value after reset

Table 10-86. Destination FIFO Destination Address Register (DFDSTn) Field Descriptions

Bit	Field	Value	Description
31-0	DADDR	0	Destination address for the destination FIFO register set. When a transfer request (TR) is complete, the final value should be the address of the last write command issued.

10.5.3.6.14 Destination FIFO B-Index Register (DFBIDXn)

The destination FIFO B-index register (DFBIDXn) is shown in Figure 10-85 and described in Table 10-87.

NOTE: The value for *n* varies from 0 to DSTREGDEPTH for the given EDMA3TC.

Figure 10-85. Destination FIFO B-Index Register (DFBIDXn)

31	16
DBIDX	
R-0	

15	0
SBIDX	
R-0	

LEGEND: R = Read only; -*n* = value after reset

Table 10-87. Destination FIFO B-Index Register (DFBIDXn) Field Descriptions

Bit	Field	Value	Description
31-16	DBIDX	0-FFFFh	B-Index offset between destination arrays. Represents the offset in bytes between the starting address of each destination.
15-0	SBIDX	0	Always read as 0.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.3.6.15 Destination FIFO Memory Protection Proxy Register (DFMPPRXY*n*)

The destination FIFO memory protection proxy register (DFMPPRXY*n*) is shown in Figure 10-86 and described in Table 10-79.

Figure 10-86. Destination FIFO Memory Protection Proxy Register (DFMPPRXY*n*)

31–16	15–9	8	7–4	3–0
Reserved	Reserved	PRIV	Reserved	PRIVID
R-0	R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 10-88. Destination FIFO Memory Protection Proxy Register (DFMPPRXY*n*) Field Descriptions

Bit	Field	Value	Description
31-9	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
8	PRIV		Privilege level. This contains the Privilege level used by the EDMA3 programmer to set up the parameter entry in the channel controller. This field is set up when the associated TR is submitted to the EDMA3TC. The privilege ID is used while issuing read and write command to the target endpoints so that the target endpoints can perform memory protection checks based on the PRIV of the host that set up the DMA transaction.
		0	User-level privilege
		1	Supervisor-level privilege
7-4	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
3-0	PRIVID	0-Fh	Privilege ID. This contains the Privilege ID of the EDMA3 programmer that set up the parameter entry in the channel controller. This field is set up when the associated TR is submitted to the EDMA3TC. This PRIVID value is used while issuing read and write commands to the target endpoints so that the target endpoints can perform memory protection checks based on the PRIVID of the host that set up the DMA transaction.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.3.6.16 Destination FIFO Count Reload Register (DFCNTRLD*n*)

The destination FIFO count reload register (DFCNTRLD*n*) is shown in Figure 10-87 and described in Table 10-89.

Figure 10-87. Destination FIFO Count Reload Register (DFCNTRLD*n*)

31		16
	Reserved	
	R-0	

15		0
	ACNTRLD	
	R-0	

LEGEND: R = Read only; -*n* = value after reset

Table 10-89. Destination FIFO Count Reload Register (DFCNTRLD*n*) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.
15-0	ACNTRLD	0-FFFFh	A-count reload value. Represents the originally programmed value of ACNT. The reload value is used to reinitialize ACNT after each array is serviced.

10.5.3.6.17 Destination FIFO Source Address B-Reference Register (DFSRCBREF*n*)

The destination FIFO source address B-reference register (DFSRCBREF*n*) is shown in Figure 10-88 and described in Table 10-90.

Figure 10-88. Destination FIFO Source Address B-Reference Register (DFSRCBREF*n*)

31		16
	Reserved	
	R-0	

15		0
	Reserved	
	R-0	

LEGEND: R = Read only; -*n* = value after reset

Table 10-90. Destination FIFO Source Address B-Reference Register (DFSRCBREF*n*) Field Descriptions

Bit	Field	Value	Description
31-0	Reserved	0	Reserved. Always read as 0.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

10.5.3.6.18 Destination FIFO Destination Address B-Reference (DFDSTBREF*n*)

The destination FIFO destination address B-reference register (DFDSTBREF*n*) is shown in Figure 10-89 and described in Table 10-91.

Figure 10-89. Destination FIFO Destination Address B-Reference Register (DFDSTBREF*n*)

31	16
DADDRBREF	
R-0	

15	0
DADDRBREF	
R-0	

LEGEND: R = Read only; -*n* = value after reset

Table 10-91. Destination FIFO Destination Address B-Reference Register (DFDSTBREF*n*) Field Descriptions

Bit	Field	Value	Description
31-0	DADDRBREF	0-FFFF FFFFh	Destination address reference for the destination FIFO register set. Represents the starting address for the array currently being written.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Chapter 11

SPRUH73–July 2011

Touchscreen Controller

This chapter describes the touchscreen controller of the device.

Topic **Page**

11.1 Introduction **1440**
11.2 Integration **1441**
11.3 Functional Description **1443**
11.4 Operational Modes **1446**
11.5 TSC_ADC_SS Registers **1447**

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.1 Introduction

The touchscreen controller and analog-to-digital converter subsystem (TSC_ADC_SS) is an 8-channel general-purpose analog-to-digital converter (ADC) with optional support for interleaving touchscreen (TS) conversions for a 4-wire, 5-wire, or 8-wire resistive panel. The TSC_ADC_SS can be configured for use in one of the following applications:

- 8 general-purpose ADC channels
- 4-wire TSC with 4 general-purpose ADC channels
- 5-wire TSC with 3 general-purpose ADC channels
- 8-wire TSC

11.1.1 TSC_ADC Features

The main features of the TSC_ADC_SS include:

- Support for 4-wire, 5-wire, and 8-wire resistive TS panels
- Support for interleaving TS capture and general-purpose ADC modes
- Programmable FSM sequencer that supports 16 steps:
  - Software register bit for start of conversion
  - Optional start of conversion HW synchronized to Pen touch or external HW event (but not both)
  - Single conversion (one-shot)
  - Continuous conversions
  - Sequence through all input channels based on a mask
  - Programmable OpenDelay before sampling each channel
  - Programmable sampling delay for each channel
  - Programmable averaging of input samples - 16/8/4/2/1
  - Differential or singled ended mode setting for each channel
  - Store data in either of two FIFO groups
  - Option to encode channel number with data
  - Support for servicing FIFOs via DMA or CPU
  - Programmable DMA Request event (for each FIFO)
  - Dynamically enable or disable channel inputs during operation
  - Stop bit to end conversion
- Support for the following interrupts and status, with masking:
  - Interrupt for HW pen (touch) event
  - Interrupt for HW Pen Up event
  - Interrupt after a sequence of conversions (all non-masked channels)
  - Interrupt for FIFO threshold levels
  - Interrupt if sampled data is out of a programmable range
  - Interrupt for FIFO overflow and underflow conditions
  - Status bit to indicate if ADC is busy converting

11.1.2 Unsupported TSC_ADC_SS Features

This device supports all TSC_ADC_SS features.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.2 Integration

Figure 11-1 shows the integration of the TSC_ADC module in the device.

Figure 11-1. TSC_ADC Integration

11.2.1 TSC_ADC Connectivity Attributes

The general connectivity attributes for the TSC_ADC module are summarized in Table 11-1.

Table 11-1. TSC_ADC Connectivity Attributes

Attributes	Type
Power domain	Wakeup Domain
Clock domain	PD_WKUP_L4_WKUP_GCLK (OCP) PD_WKUP_ADC_FCLK (Func)
Reset signals	WKUP_DOM_RST_N
Idle/Wakeup signals	Smart idle Wakeup
Interrupt request	1 interrupt to MPU Subsystem, PRUSS and WakeM3
DMA request	2 Events
Physical address	L3 Slow slave port (DMA) L4 Wkup slave port (MMR)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.2.2 TSC_ADC Clock and Reset Management

The TSC_ADC has two clock domains. The ADC uses the adc_clk. The bus interfaces, FIFOs, sequencer, and all other lofic use the ocp_clk.

Table 11-2. TSC_ADC Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
ocp_clk OCP / Functional clock	100 MHz	CORE_CLKOUTM4 / 2	pd_pwkup_l4_wkup_gclk From PRCM
adc_clk ADC clock	24 MHz (typ)	CLK_M_OSC	pd_wkup_adc_fclk From PRCM

11.2.3 TSC_ADC Pin List

The TSC_ADC external interface signals are shown in Table 11-3.

Table 11-3. TSC_ADC Pin List

Pin	Type	Description
AN[7:0]	I	Analog Input

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.3 Functional Description

Before enabling the TSC_ADC_SS module, the user must first program the Step Configuration registers in order to configure a channel input to be sampled. There are 16 programmable Step Configuration registers which are used by the sequencer to control which switches to turn on or off (inputs to the AFE), which channel to sample, , and which mode to use (HW triggered or SW enabled, one-shot or continuous, averaging, where to save the FIFO data, etc).

11.3.1 HW Synchronized or SW Channels

The user can control the start behavior of each channel input by deciding if a channel should be sampled immediately (SW enabled) after it is enabled, or if the channel should wait for a HW event to occur first (a HW event must either be mapped to the touch screen PEN event or mapped to theHW event input signal, but not both) . Each channel input can be configured independently via the Step Configuration register.

11.3.2 Open Delay and Sample Delay

The user can program the delay between driving the inputs to the AFE and the time to send the start of conversion signal. This delay can be used to allow the voltages to stabilize on the touch screen panel before sampling. This delay is called “open delay” and can also be programmed to zero. The user also has control of the sampling time (width of the start of conversion signal) to the AFE which is called the “sample delay”. Each channel input is configured independently via the Step Delay register.

11.3.3 Averaging of Samples (1, 2, 4, 8, and 16)

Each channel input has the capability to average the sampled data. The valid averaging options are 1 (no average), 2, 4, 8, and 16. If averaging is turned on, then the channel is immediately sampled again (up to 16 times) and final averaged sample data is stored in the FIFO. Each channel input can be independently configured via the Step Configuration registers.

11.3.4 One-Shot (Single) or Continuous Mode

When the sequencer finishes cycling through all the enabled steps , the user can decide if the sequencer should stop (one-shot), or loop back and schedule a the step again (continuous).

If one-shot mode is selected, the sequencer will take care of disabling the step enable bit after the conversion. If continuous mode is selected, it is the software’s responsibility to turn off the step enable bit.

11.3.5 Interrupts

The following interrupts are supported via enable bits, and also are maskable:

- HW_PEN down interrupt is generated when the user touches the screen
- PEN_UP interrupt is generated after the sequencer detects pen is lifted
- END_OF_SEQUENCE interrupt is generated after the last active step
- Each FIFO has a underflow or overrun interrupt
- Each FIFO has a programmable threshold interrupt
- An OUT_OF_RANGE interrupt is generated if sampled data is greater than programmable value, or less than a programmable value

See the *Interrupts* chapter for more information.

11.3.6 DMA Requests

Each FIFO group can be serviced by either a DMA or by the CPU. To generate DMA requests, the user must first enable the TSC_ADC_SS_DMAENABLE_SET Register. Also, the user can program the desired number of words to generate a DMA request using the TSC_ADC_SS_DMAREQ register. When the FIFO level reaches or exceeds that value, a DMA request is generated.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

The DMA slave port allows for burst reads in order to effectively move the FIFO data. Internally, the OCP DMA address (MSB) is decoded for either FIFO 0 or FIFO 1. The lower bits of the DMA addresses are ignored since the FIFO pointers are incremented internally. Refer to **section 5** (DMA) for more information.

11.3.7 AFE Functional Block Diagram

The AFE features are listed below, and some are controlled by the TSC_ADC_SS:

- ADC sampled data is 12-bit wide
- Sampling rate can be as fast as every 15 ADC clock cycles
- Support for internal ADC clock divider logic
- Support for configuring the delay between samples also the sampling time

Figure 11-2. Functional Block Diagram

Table 11-4. AFE I/O List

Signal	Type	Function
XPPSW	In	X+ driver switch. Logic ‘1’ connects X+ to VDDA
XNNSW	In	X- driver switch. Logic ‘1’ connects X- to VSSA
YPPSW	In	Y+ driver switch. Logic ‘1’ connects Y+ to VDDA
YNNSW	In	Y- driver switch. Logic ‘1’ connects Y- to VSSA
XNPSW	In	X- driver switch. Logic ‘1’ connects X- to VDDA
YPNSW	In	Y+ driver switch. Logic ‘1’ connects Y+ to VSSA
WPNSW	In	Wiper driver switch. Logic ‘1’ connects Wiper to VSSA
SEL_RFP[2:0]	In	Multiplexer control to select the ADC positive reference voltage
SEL_RFM[1:0]	In	Multiplexer control to select the ADC negative reference voltage
SEL_INP[3:0]	In	Multiplexer control to select the ADC positive analog input
SEL_INM[3:0]	In	Multiplexer control to select the ADC negative analog input
BIAS_SEL	In	Analog Bias Select. Logic low selects internal bias
PD	In	ADC Power Down. Active high.
DIFF_CNTL	In	Set the ADC in single ended or differential mode 0:: single 1: differential mode
PENIRQ[1:0]	Out	Touch Screen interruption request signal for Wiper or X+, respectively
PENCTR[1:0]	In	Interrupt enable control pin for Wiper or X+, respectively
CLK	In	ADC input clock
SOC	In	Start Of Conversion signal. Falling edge triggers start.
EOC	Out	End Of Conversion signal. Active high indicates valid data
ADCOUT[11:0]	Out	Digital Output of ADC
IN_SPARE[3:0]	In*	For Debug purposes
OUT_SPARE[3:0]	Out	4 spare outputs

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 11-4. AFE I/O List (continued)

Signal	Type	Function
ADCREFM	In	ADC negative reference input
ADCREFP	In	ADC positive reference input
INTREF	In	ADC internal reference
AN0 / XPUL	In	Analog input 1
AN1 / XNUR	In	Analog input 2
AN 2/ YPLL	In	Analog input 3
AN 3/ YNLR	In	Analog input 4
AN 4 / Wiper	In	Analog input 5
AN 5	In	Analog input 6
AN 6	In	Analog input 7
AN 7	In	Analog input 8
HHV	In	HHV is high when core VDD is inactive. VDDA domain, also powers down ADC when 1
REXT_BIAS	In	External bias resistor

See the AFE functional spec for more detail (hardware reset values for inputs).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.4 Operational Modes

The sequencer is completely controlled by software and behaves accordingly to how the **Step Registers** are programmed. A **step** is the general term for sampling a channel input. It is defined by the programmer who decides which input values to send to the AFE (via the StepConfig register), and also how (via StepConfig register) and when (via StepDelay register) to sample a channel input.

A **step** consists of these three registers:

- **StepEnable** – enables or disables the step
- **StepConfig** - controls the input values to the ADC (the reference voltages, the pull up/down transistor biasing, which input channel to sample, differential control, HW synchronized or SW enabled, averaging, and which FIFO group to save the data)
- **StepDelay** - controls the OpenDelay (the time between driving the AFE inputs until sending the SOC signal to the AFE), and also controls the SampleDelay (the time between the rising edge of the start signal and the falling edge of the start signal)

The sequencer supports a total of 16 programmable steps, a Touchscreen Charge step, and an Idle step. Each step consists of the three registers above (StepEnable, StepConfig, and StepDelay), except the Idle step does not have a StepEnable bit (it must always be on), and it also does not have a StepDelay register. Also, during the Idle and Touch Screen Charge steps, the sequencer does not actually sample a channel after applying the StepConfiguration values.

Assuming all the steps are were configured as general purpose mode (no touchscreen), then the steps would be configured as SW enabled. When the TSC_ADC_SS is first enabled, the sequencer will always start in the Idle step and then wait for a StepEnable[n] bit to turn on. After a step is enabled, the sequencer will start with the lowest step (1) and continue until step (16),. If a step is not enabled, then sequencer will skip to the next step. If all steps are disabled, then the sequencer will remain in the IDLE state and continue to apply the Idle StepConfig settings.

Assuming a touch screen only mode (no general purpose channels) the steps could be configured as HW synchronized triggered (mapped to the PEN event).. The sequencer would wait in the IDLE state until a HW pen down event occurred and then begin the HW step conversions. The Touch Screen Charge step occurs after the last HW step before going back to the Idle state (the TS Charge step is needed to charge touch screen capacitance which allows the controller to detect subsequent Pen touch events).

Assuming a mixed mode application (touch screen and general purpose channels), the user can configure the steps as either HW triggered (mapped to Pen event) or SW enabled. If the sequencer is in the IDLE state and a HW pen event occurs, then the HW steps (from lowest to highest) are always scheduled first, followed by the TS Charge step. If there is no HW event, then the SW enabled steps are scheduled instead

If a HW event occurs while the sequencer is in the middle of scheduling the SW steps, the user can program the scheduler to allow preemption. If the HW preempt control bit is enabled, the sequenceer will allow the current SW step to finish and then schedule the HW steps. After the last HW step and TS Charge step are completed, the sequencer will continue from the next SW step (before the preemption occurred). If the HW preemption is disabled, then the touch event will be ignored until the last software step is completed; if the touch event is removed before the last software step is finished, then the touch event will be missed.

Even if a touch screen is not present, the user can still configure the steps to be HW synchronize by mapping to the HW event input signal. This HW event input signal could be driven at the SOC level from a timer interrupt or some other IP and can be used to start a channel conversion.

When mapping is set for the input HW event signal, then the TSC_ADC_SS will wait for a rising edge transition (from low to high) before starting. The input HW event signal is captured on the internal L4 OCP clock domain. The HW event input signal should be held for at least 2 TSC_ADC_SS OCP clocks (L4 frequency).

An END_OF_SEQUENCE interrupt and the PEN_UP interrupt are generated after the last active step is completed before going back to the IDLE state. The END_OF_SEQUENCE interrupt does not mean data is in the FIFO (should use the FIFO interrupts and word count reg)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.4.1 PenCtrl and PenIRQ

The Pen IRQ can only occur if the correct Pen Ctrl bits are high (AFE Pen Ctrl bits 6:5 in the TSC_ADC control register) and if the correct ground transistor biasing is set in the StepConfig [N] Register. Refer to the application notes for the correct settings.

If a step is configured as HW synchronized, the sequencer will override the PenCtrl bits set by the software (bits 6 and 5) once it transitions from the IdleStep. The sequencer will automatically mask the PenCtrl bits (override them and turn them off) so that the ADC can get an accurate measurement from the x and y channels. After the last HW synchronized step, the sequence will go to the TS Charge step and the pen_overide mask is removed and the values set by the SW (bits 6:5) will have control. The pen_irq will be temporarily ignored during the TS Charge step (Hw will mask any potential glitchs that may occur)

If the sequencer is not using the HW synchronized approach, (all the steps are configured as software enabled), then it is the software programmer's responsibility to correctly turn on and off the pen ctrl bits to the AFE in order get the correct measures from the touch screen. It is also software job to send the charge step and ignore any potential glitches (by using pen_irq status register).

It is also possible to detect the HW Pen event even if all the StepEnable[n] bits are off. By setting the PenCtrl bit to 1, and configuring the IdleStep Configuration register to correctly bias the transistor to grnd, the HW_PEN event will be generated. The flowchart for the sequencer is shown in Figure 11-3 and an example timing diagram in Figure 11-4.

Figure 11-3. Sequencer FSM

The previous diagram does not actually represent clock cycles but instead illustrates how the scheduler will work(however, each shaded box above does represent a FSM state and will use a clock cycle). Using the minimum delay values, the ADC can sample at 15 ADC clocks per sample. Below is an example timing diagram illustrating the states of the sequencer and also the showing when the StepConfig and the StepDelay registers values are applied. The below example assumes the steps are software controlled, and averaging is turned off.

Figure 11-4. Example Timing Diagram for Sequencer

The Idle StepConfig is always enabled and applied when the FSM is in the IDLE state (i.e, either waiting for a HW event or waiting for a step to be enabled). The Idle StepConfig can not be disabled.

Once the TSC_ADC_SS is enabled and assuming at least one StepEnable[N] is active, the FSM will transition from the Idle state and apply the first active StepConfig[N] and StepDelay[N] register settings. It is possible for the OpenDelay[N] value to be 0, and therefore the FSM will immediately skip to the SampleDelay[N] state (which is minimum 1 clock cycle). The ADC will begin the sampling on the falling edge of the SOC signal. After the ADC is finished converting the channel data (13 cycles later), the EOC signal is sent and the FSM will then apply the next active step [N+1].

This process is repeated and continued (from step 1 to step 16) until the last active step is completed.

11.5 TSC_ADC_SS Registers

Table 11-5 lists the memory-mapped registers for the TSC_ADC_SS. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 11-5 should be considered as reserved locations and the register contents should not be modified.

Table 11-5. TSC_ADC_SS REGISTERS

Offset	Acronym	Register Description	Section
0h	REVISION	Revision Register	Section 11.5.1
64h	STEPCONFIG4	Step Configuration 4	Section 11.5.2
68h	STEPDELAY4	Step Delay Register 4	Section 11.5.3
6Ch	STEPCONFIG5	Step Configuration 5	Section 11.5.4
70h	STEPDELAY5	Step Delay Register 5	Section 11.5.5
74h	STEPCONFIG6	Step Configuration 6	Section 11.5.6

Copyright © 2011, Texas Instruments Incorporated

Table 11-5. TSC_ADC_SS REGISTERS (continued)

Offset	Acronym	Register Description	Section
Ch	IRQSTATUS_RAW	IRQ status (unmasked)	Section 12.5.29
78h	STEPDELAY6	Step Delay Register 6	Section 11.5.8
7Ch	STEPCONFIG7	Step Configuration 7	Section 11.5.9
80h	STEPDELAY7	Step Delay Register 7	Section 11.5.10
84h	STEPCONFIG8	Step Configuration 8	Section 11.5.11
88h	STEPDELAY8	Step Delay Register 8	Section 11.5.12
8Ch	STEPCONFIG9	Step Configuration 9	Section 11.5.13
90h	STEPDELAY9	Step Delay Register 9	Section 11.5.14
94h	STEPCONFIG10	Step Configuration 10	Section 11.5.15
98h	STEPDELAY10	Step Delay Register 10	Section 11.5.16
9Ch	STEPCONFIG11	Step Configuration 11	Section 11.5.17
10h	IRQSTATUS	IRQ status (masked)	Section 12.5.30
A0h	STEPDELAY11	Step Delay Register 11	Section 11.5.19
A4h	STEPCONFIG12	Step Configuration 12	Section 11.5.20
A8h	STEPDELAY12	Step Delay Register 12	Section 11.5.21
ACh	STEPCONFIG13	Step Configuration 13	Section 11.5.22
B0h	STEPDELAY13	Step Delay Register 13	Section 11.5.23
B4h	STEPCONFIG14	Step Configuration 14	Section 11.5.24
B8h	STEPDELAY14	Step Delay Register 14	Section 11.5.25
BCh	STEPCONFIG15	Step Configuration 15	Section 11.5.26
C0h	STEPDELAY15	Step Delay Register 15	Section 11.5.27
C4h	STEPCONFIG16	Step Configuration 16	Section 11.5.28
14h	IRQENABLE_SET	IRQ enable set bits	Section 12.5.31
C8h	STEPDELAY16	Step Delay Register 16	Section 11.5.30
CCh	FIFO0COUNT	FIFO0 word count@TSC_ADC_SS_FIFO0 Word Count Register	Section 11.5.31
D0h	FIFO0THRESHOLD	FIFO0 Threshold trigger@TSC_ADC_SS_FIFO0 Threshold Level Register	Section 11.5.32
D4h	DMA0REQ	FIFO0 DMA req0 trigger@TSC_ADC_SS_FIFO0 DMA REQUEST Register	Section 11.5.33
D8h	FIFO1COUNT	FIFO1 word count@TSC_ADC_SS_FIFO1 Word Count Register	Section 11.5.34
DCh	FIFO1THRESHOLD	FIFO1 Threshold trigger@TSC_ADC_SS_FIFO1 Threshold Level Register	Section 11.5.35
E0h	DMA1REQ	FIFO1 DMA req1 trigger@TSC_ADC_SS_FIFO1 DMA Request Register	Section 11.5.36
E4h	FIFO0DATA	ADC_ FIFO0 _READ Data @TSC_ADC_SS_FIFO0 READ Register	Section 11.5.37
E8h	FIFO1DATA	ADC FIFO1_READ Data@TSC_ADC_SS_FIFO1 READ Register	Section 11.5.38
18h	IRQENABLE_CLR	IRQ enable clear bits	Section 11.5.39
1Ch	IRQWAKEUP	IRQ wakeup enable	Section 11.5.40
20h	DMAENABLE_SET	Per-Line DMA set	Section 11.5.41
24h	DMAENABLE_CLR	Per-Line DMA clr	Section 11.5.42
4h	SYSCONFIG	SysConfig Register	Section 16.2.2.2
28h	CTRL	@TSC_ADC_SS Control Register	Section 12.5.16
2Ch	ADCSTAT	General Status bits @TSC_ADC_SS_Sequencer_Status Register	Section 11.5.45
30h	ADCRANGE	High and Low Range Threshold@TSC_ADC_SS_Range_Check Register	Section 11.5.46

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 11-5. TSC_ADC_SS REGISTERS (continued)

Offset	Acronym	Register Description	Section
34h	ADC_CLKDIV	ADC clock divider register@TSC_ADC_SS_Clock_Divider Register	Section 11.5.47
38h	ADC_MISC	AFE misc debug@TSC_ADC_SS_MISC Register	Section 11.5.48
3Ch	STEPENABLE	Step Enable	Section 11.5.49
40h	IDLECONFIG	Idle Step configuration@TSC_ADC_SS_IDLE_StepConfig Register	Section 11.5.50
44h	TS_CHARGE_STEPCONFIG	TS Charge StepConfiguration@TSC_ADC_SS_TS_Charge_StepConfig Register	Section 11.5.51
48h	TS_CHARGE_DELAY	TS Charge Delay Register@TSC_ADC_SS_TS_Charge_StepDelay Register	Section 11.5.52
4Ch	STEPCONFIG1	Step Configuration 1	Section 11.5.53
8h	IRQ_EOI	IRQ EOI	Section 11.5.54
50h	STEPDELAY1	Step Delay Register 1	Section 11.5.55
54h	STEPCONFIG2	Step Configuration 2	Section 11.5.56
58h	STEPDELAY2	Step Delay Register 2	Section 11.5.57
5Ch	STEPCONFIG3	Step Configuration 3	Section 11.5.58
60h	STEPDELAY3	Step Delay Register 3	Section 11.5.59

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.1 REVISION Register (offset = 0h) [reset = 47300001h]

Revision Register

REVISION is shown in Figure 11-5 and described in Table 11-6.

Figure 11-5. REVISION Register

31	30	29	28	27	26	25	24
SCHEME		Reserved		FUNC			
R-1h		R-0h		R-730h			

23	22	21	20	19	18	17	16
FUNC							
R-730h							

15	14	13	12	11	10	9	8
R_RTL					X_MAJOR		
R-0h					R-0h		

7	6	5	4	3	2	1	0
CUSTOM		Y_MINOR					
R-0h		R-1h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-6. REVISION Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	HL 0.8 scheme
29-28	Reserved	R	0h	Always read as 0. Writes have no affect.
27-16	FUNC	R	730h	Functional Number
15-11	R_RTL	R	0h	RTL revision. Will vary depending on release.
10-8	X_MAJOR	R	0h	Major revision.
7-6	CUSTOM	R	0h	Custom revision.
5-0	Y_MINOR	R	1h	Minor revision

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.2 STEPCONFIG4 Register (offset = 64h) [reset = 0h]

Step Configuration 4

STEPCONFIG4 is shown in Figure 11-6 and described in Table 11-7.

Figure 11-6. STEPCONFIG4 Register

31	30	29	28	27	26	25	24
Reserved				Range_check	FIFO_select	Diff_CNTRL	SEL_RFM_SWC_1_0
-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
SEL_RFM_SWC_1_0	SEL_INP_SWC_3_0				SEL_INM_SWM3_0		
R/W-0h	R/W-0h				R/W-0h		

15	14	13	12	11	10	9	8
SEL_INM_SWM3_0	SEL_RFP_SWC_2_0			WPNSW_SWC	YPNSW_SWC	XNPSW_SWC	YNNSW_SWC
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
YPPSW_SWC	XNNSW_SWC	XPPSW_SWC	Averaging			Mode	
R/W-0h	R/W-0h	R/W-0h	R/W-0h			R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-7. STEPCONFIG4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved		0h	
27	Range_check	R/W	0h	0 disable out of range check 1 - Compare adc data with range check register
26	FIFO_select	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 01 -> FIFO 1
25	Diff_CNTRL	R/W	0h	Differential Control Pin
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins SW configuration
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins SW configuration
18-15	SEL_INM_SWM3_0	R/W	0h	SEL_INP pins for neg differential
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins SW configuration
11	WPNSW_SWC	R/W	0h	WPNSW pin SW configuration
10	YPNSW_SWC	R/W	0h	YPNSW pin SW configuration
9	XNPSW_SWC	R/W	0h	XNPSW pin SW configuration
8	YNNSW_SWC	R/W	0h	YNNSW pin SW configuration
7	YPPSW_SWC	R/W	0h	YPPSW pin SW configuration
6	XNNSW_SWC	R/W	0h	XNNSW pin SW configuration
5	XPPSW_SWC	R/W	0h	XPPSW pin SW configuration
4-2	Averaging	R/W	0h	Number of samplings to average: 000 -> no average001 2 samples average010 4 samples average011 8 samples average 100 16 samples average
1-0	Mode	R/W	0h	00 SW enabled, one-shot01 SW enabled, continuous 10 HW synchronized, one-shot11 HW synchronized, continuous

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.3 STEPDELAY4 Register (offset = 68h) [reset = 0h]

Step Delay Register 4

STEPDELAY4 is shown in Figure 11-7 and described in Table 11-8.

Figure 11-7. STEPDELAY4 Register

31	30	29	28	27	26	25	24
SampleDelay							
R/W-0h							

23	22	21	20	19	18	17	16
Reserved						OpenDelay	
-0h						R/W-0h	

15	14	13	12	11	10	9	8
OpenDelay							
R/W-0h							

7	6	5	4	3	2	1	0
OpenDelay							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-8. STEPDELAY4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SampleDelay	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SOC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	Reserved		0h	
17-0	OpenDelay	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.4 STEPCONFIG5 Register (offset = 6Ch) [reset = 0h]

Step Configuration 5

STEPCONFIG5 is shown in Figure 11-8 and described in Table 11-9.

Figure 11-8. STEPCONFIG5 Register

31	30	29	28	27	26	25	24
Reserved				Range_check	FIFO_select	Diff_CNTRL	SEL_RFM_SWC_1_0
-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
SEL_RFM_SWC_1_0	SEL_INP_SWC_3_0				SEL_INM_SWM3_0		
R/W-0h	R/W-0h				R/W-0h		

15	14	13	12	11	10	9	8
SEL_INM_SWM3_0	SEL_RFP_SWC_2_0			WPNSW_SWC	YPNSW_SWC	XNPSW_SWC	YNNSW_SWC
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
YPPSW_SWC	XNNSW_SWC	XPPSW_SWC	Averaging			Mode	
R/W-0h	R/W-0h	R/W-0h	R/W-0h			R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-9. STEPCONFIG5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved		0h	
27	Range_check	R/W	0h	0 disable out of range check 1 - Compare adc data with range check register
26	FIFO_select	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 01 -> FIFO 1
25	Diff_CNTRL	R/W	0h	Differential Control Pin
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins SW configuration
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins SW configuration
18-15	SEL_INM_SWM3_0	R/W	0h	SEL_INP pins for neg differential
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins SW configuration
11	WPNSW_SWC	R/W	0h	WPNSW pin SW configuration
10	YPNSW_SWC	R/W	0h	YPNSW pin SW configuration
9	XNPSW_SWC	R/W	0h	XNPSW pin SW configuration
8	YNNSW_SWC	R/W	0h	YNNSW pin SW configuration
7	YPPSW_SWC	R/W	0h	YPPSW pin SW configuration
6	XNNSW_SWC	R/W	0h	XNNSW pin SW configuration
5	XPPSW_SWC	R/W	0h	XPPSW pin SW configuration
4-2	Averaging	R/W	0h	Number of samplings to average: 000 -> no average001 2 samples average010 4 samples average011 8 samples average 100 16 samples average
1-0	Mode	R/W	0h	00 SW enabled, one-shot01 SW enabled, continuous 10 HW synchronized, one-shot11 HW synchronized, continuous

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.5 STEPDELAY5 Register (offset = 70h) [reset = 0h]

Step Delay Register 5

STEPDELAY5 is shown in Figure 11-9 and described in Table 11-10.

Figure 11-9. STEPDELAY5 Register

31	30	29	28	27	26	25	24
SampleDelay							
R/W-0h							

23	22	21	20	19	18	17	16
Reserved						OpenDelay	
-0h						R/W-0h	

15	14	13	12	11	10	9	8
OpenDelay							
R/W-0h							

7	6	5	4	3	2	1	0
OpenDelay							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-10. STEPDELAY5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SampleDelay	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SOC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	Reserved		0h	
17-0	OpenDelay	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.6 STEPCONFIG6 Register (offset = 74h) [reset = 0h]

Step Configuration 6

STEPCONFIG6 is shown in Figure 11-10 and described in Table 11-11.

Figure 11-10. STEPCONFIG6 Register

31	30	29	28	27	26	25	24
Reserved				Range_check	FIFO_select	Diff_CNTRL	SEL_RFM_SWC_1_0
-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
SEL_RFM_SWC_1_0	SEL_INP_SWC_3_0				SEL_INM_SWM3_0		
R/W-0h	R/W-0h				R/W-0h		

15	14	13	12	11	10	9	8
SEL_INM_SWM3_0	SEL_RFP_SWC_2_0			WPNSW_SWC	YPNSW_SWC	XNPSW_SWC	YNNSW_SWC
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
YPPSW_SWC	XNNSW_SWC	XPPSW_SWC	Averaging			Mode	
R/W-0h	R/W-0h	R/W-0h	R/W-0h			R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-11. STEPCONFIG6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved		0h	
27	Range_check	R/W	0h	0 disable out of range check 1 - Compare adc data with range check register
26	FIFO_select	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 01 -> FIFO 1
25	Diff_CNTRL	R/W	0h	Differential Control Pin
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins SW configuration
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins SW configuration
18-15	SEL_INM_SWM3_0	R/W	0h	SEL_INP pins for neg differential
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins SW configuration
11	WPNSW_SWC	R/W	0h	WPNSW pin SW configuration
10	YPNSW_SWC	R/W	0h	YPNSW pin SW configuration
9	XNPSW_SWC	R/W	0h	XNPSW pin SW configuration
8	YNNSW_SWC	R/W	0h	YNNSW pin SW configuration
7	YPPSW_SWC	R/W	0h	YPPSW pin SW configuration
6	XNNSW_SWC	R/W	0h	XNNSW pin SW configuration
5	XPPSW_SWC	R/W	0h	XPPSW pin SW configuration
4-2	Averaging	R/W	0h	Number of samplings to average: 000 -> no average001 2 samples average010 4 samples average011 8 samples average 100 16 samples average
1-0	Mode	R/W	0h	00 SW enabled, one-shot01 SW enabled, continuous 10 HW synchronized, one-shot11 HW synchronized, continuous

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.7 IRQSTATUS_RAW Register (offset = Ch) [reset = 0h]

IRQ status (unmasked)

IRQSTATUS_RAW is shown in Figure 12-54 and described in Table 12-41.

Figure 11-11. IRQSTATUS_RAW Register

31	30	29	28	27	26	25	24
Reserved							
-0h							

23	22	21	20	19	18	17	16
Reserved							
-0h							

15	14	13	12	11	10	9	8
Reserved					PEN_IRQ_synchronized	Pen_Up_Event	Out_of_Range
-0h					R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
FIFO1_Underflow	FIFO1_Overrun	FIFO1_Threshold	FIFO0_Underflow	FIFO0_Overrun	FIFO0_Threshold	End_of_Sequence	HW_Pen_Event_asynchronous
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 11-12. IRQSTATUS_RAW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	Reserved		0h	
10	PEN_IRQ_synchronized	R/W	0h	Write 0: No action Write 1: Set event (debug) Read 0: No event pending Read 1: Event pending
9	Pen_Up_Event	R/W	0h	Write 0: No action Write 1: Set event (debug) Read 0: No event pending Read 1: Event pending
8	Out_of_Range	R/W	0h	Write 0: No action Write 1: Set event (debug) Read 0: No event pending Read 1: Event pending
7	FIFO1_Underflow	R/W	0h	Write 0: No action Write 1: Set event (debug) Read 0: No event pending Read 1: Event pending
6	FIFO1_Overrun	R/W	0h	Write 0: No action Write 1: Set event (debug) Read 0: No event pending Read 1: Event pending
5	FIFO1_Threshold	R/W	0h	Write 0: No action Write 1: Set event (debug) Read 0: No event pending Read 1: Event pending
4	FIFO0_Underflow	R/W	0h	Write 0: No action Write 1: Set event (debug) Read 0: No event pending Read 1: Event pending
3	FIFO0_Overrun	R/W	0h	Write 0: No action Write 1: Set event (debug) Read 0: No event pending Read 1: Event pending

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 11-12. IRQSTATUS_RAW Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	FIFO0_Threshold	R/W	0h	Write 0: No action Write 1: Set event (debug) Read 0: No event pending Read 1: Event pending
1	End_of_Sequence	R/W	0h	Write 0: No action Write 1: Set event (debug) Read 0: No event pending Read 1: Event pending
0	HW_Pen_Event_asynchronous	R/W	0h	Write 0: No action Write 1: Set event (debug) Read 0: No event pending Read 1: Event pending

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.8 STEPDELAY6 Register (offset = 78h) [reset = 0h]

Step Delay Register 6

STEPDELAY6 is shown in Figure 11-12 and described in Table 11-13.

Figure 11-12. STEPDELAY6 Register

31	30	29	28	27	26	25	24
SampleDelay							
R/W-0h							

23	22	21	20	19	18	17	16
Reserved						OpenDelay	
-0h						R/W-0h	

15	14	13	12	11	10	9	8
OpenDelay							
R/W-0h							

7	6	5	4	3	2	1	0
OpenDelay							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-13. STEPDELAY6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SampleDelay	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SOC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	Reserved		0h	
17-0	OpenDelay	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.9 STEPCONFIG7 Register (offset = 7Ch) [reset = 0h]

Step Configuration 7

STEPCONFIG7 is shown in Figure 11-13 and described in Table 11-14.

Figure 11-13. STEPCONFIG7 Register

31	30	29	28	27	26	25	24
Reserved				Range_check	FIFO_select	Diff_CNTRL	SEL_RFM_SWC_1_0
-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
SEL_RFM_SWC_1_0	SEL_INP_SWC_3_0				SEL_INM_SWM3_0		
R/W-0h	R/W-0h				R/W-0h		

15	14	13	12	11	10	9	8
SEL_INM_SWM3_0	SEL_RFP_SWC_2_0			WPNSW_SWC	YPNSW_SWC	XNPSW_SWC	YNNSW_SWC
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
YPPSW_SWC	XNNSW_SWC	XPPSW_SWC	Averaging			Mode	
R/W-0h	R/W-0h	R/W-0h	R/W-0h			R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-14. STEPCONFIG7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved		0h	
27	Range_check	R/W	0h	0 disable out of range check 1 - Compare adc data with range check register
26	FIFO_select	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 01 -> FIFO 1
25	Diff_CNTRL	R/W	0h	Differential Control Pin
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins SW configuration
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins SW configuration
18-15	SEL_INM_SWM3_0	R/W	0h	SEL_INP pins for neg differential
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins SW configuration
11	WPNSW_SWC	R/W	0h	WPNSW pin SW configuration
10	YPNSW_SWC	R/W	0h	YPNSW pin SW configuration
9	XNPSW_SWC	R/W	0h	XNPSW pin SW configuration
8	YNNSW_SWC	R/W	0h	YNNSW pin SW configuration
7	YPPSW_SWC	R/W	0h	YPPSW pin SW configuration
6	XNNSW_SWC	R/W	0h	XNNSW pin SW configuration
5	XPPSW_SWC	R/W	0h	XPPSW pin SW configuration
4-2	Averaging	R/W	0h	Number of samplings to average: 000 -> no average001 2 samples average010 4 samples average011 8 samples average 100 16 samples average
1-0	Mode	R/W	0h	00 SW enabled, one-shot01 SW enabled, continuous 10 HW synchronized, one-shot11 HW synchronized, continuous

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.10 STEPDELAY7 Register (offset = 80h) [reset = 0h]

Step Delay Register 7

STEPDELAY7 is shown in Figure 11-14 and described in Table 11-15.

Figure 11-14. STEPDELAY7 Register

31	30	29	28	27	26	25	24
SampleDelay							
R/W-0h							

23	22	21	20	19	18	17	16
Reserved						OpenDelay	
-0h						R/W-0h	

15	14	13	12	11	10	9	8
OpenDelay							
R/W-0h							

7	6	5	4	3	2	1	0
OpenDelay							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-15. STEPDELAY7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SampleDelay	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SOC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	Reserved		0h	
17-0	OpenDelay	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.11 STEPCONFIG8 Register (offset = 84h) [reset = 0h]

Step Configuration 8

STEPCONFIG8 is shown in Figure 11-15 and described in Table 11-16.

Figure 11-15. STEPCONFIG8 Register

31	30	29	28	27	26	25	24
Reserved				Range_check	FIFO_select	Diff_CNTRL	SEL_RFM_SWC_1_0
-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
SEL_RFM_SWC_1_0	SEL_INP_SWC_3_0				SEL_INM_SWM3_0		
R/W-0h	R/W-0h				R/W-0h		

15	14	13	12	11	10	9	8
SEL_INM_SWM3_0	SEL_RFP_SWC_2_0			WPNSW_SWC	YPNSW_SWC	XNPSW_SWC	YNNSW_SWC
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
YPPSW_SWC	XNNSW_SWC	XPPSW_SWC	Averaging			Mode	
R/W-0h	R/W-0h	R/W-0h	R/W-0h			R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-16. STEPCONFIG8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved		0h	
27	Range_check	R/W	0h	0 disable out of range check 1 - Compare adc data with range check register
26	FIFO_select	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 01 -> FIFO 1
25	Diff_CNTRL	R/W	0h	Differential Control Pin
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins SW configuration
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins SW configuration
18-15	SEL_INM_SWM3_0	R/W	0h	SEL_INP pins for neg differential
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins SW configuration
11	WPNSW_SWC	R/W	0h	WPNSW pin SW configuration
10	YPNSW_SWC	R/W	0h	YPNSW pin SW configuration
9	XNPSW_SWC	R/W	0h	XNPSW pin SW configuration
8	YNNSW_SWC	R/W	0h	YNNSW pin SW configuration
7	YPPSW_SWC	R/W	0h	YPPSW pin SW configuration
6	XNNSW_SWC	R/W	0h	XNNSW pin SW configuration
5	XPPSW_SWC	R/W	0h	XPPSW pin SW configuration
4-2	Averaging	R/W	0h	Number of samplings to average: 000 -> no average001 2 samples average010 4 samples average011 8 samples average 100 16 samples average
1-0	Mode	R/W	0h	00 SW enabled, one-shot01 SW enabled, continuous 10 HW synchronized, one-shot11 HW synchronized, continuous

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.12 STEPDELAY8 Register (offset = 88h) [reset = 0h]

Step Delay Register 8

STEPDELAY8 is shown in Figure 11-16 and described in Table 11-17.

Figure 11-16. STEPDELAY8 Register

31	30	29	28	27	26	25	24
SampleDelay							
R/W-0h							

23	22	21	20	19	18	17	16
Reserved						OpenDelay	
-0h						R/W-0h	

15	14	13	12	11	10	9	8
OpenDelay							
R/W-0h							

7	6	5	4	3	2	1	0
OpenDelay							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-17. STEPDELAY8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SampleDelay	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SOC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	Reserved		0h	
17-0	OpenDelay	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.13 STEPCONFIG9 Register (offset = 8Ch) [reset = 0h]

Step Configuration 9

STEPCONFIG9 is shown in Figure 11-17 and described in Table 11-18.

Figure 11-17. STEPCONFIG9 Register

31	30	29	28	27	26	25	24
Reserved				Range_check	FIFO_select	Diff_CNTRL	SEL_RFM_SWC_1_0
-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
SEL_RFM_SWC_1_0	SEL_INP_SWC_3_0				SEL_INM_SWM3_0		
R/W-0h	R/W-0h				R/W-0h		

15	14	13	12	11	10	9	8
SEL_INM_SWM3_0	SEL_RFP_SWC_2_0			WPNSW_SWC	YPNSW_SWC	XNPSW_SWC	YNNSW_SWC
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
YPPSW_SWC	XNNSW_SWC	XPPSW_SWC	Averaging			Mode	
R/W-0h	R/W-0h	R/W-0h	R/W-0h			R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-18. STEPCONFIG9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved		0h	
27	Range_check	R/W	0h	0 disable out of range check 1 - Compare adc data with range check register
26	FIFO_select	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 01 -> FIFO 1
25	Diff_CNTRL	R/W	0h	Differential Control Pin
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins SW configuration
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins SW configuration
18-15	SEL_INM_SWM3_0	R/W	0h	SEL_INP pins for neg differential
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins SW configuration
11	WPNSW_SWC	R/W	0h	WPNSW pin SW configuration
10	YPNSW_SWC	R/W	0h	YPNSW pin SW configuration
9	XNPSW_SWC	R/W	0h	XNPSW pin SW configuration
8	YNNSW_SWC	R/W	0h	YNNSW pin SW configuration
7	YPPSW_SWC	R/W	0h	YPPSW pin SW configuration
6	XNNSW_SWC	R/W	0h	XNNSW pin SW configuration
5	XPPSW_SWC	R/W	0h	XPPSW pin SW configuration
4-2	Averaging	R/W	0h	Number of samplings to average: 000 -> no average001 2 samples average010 4 samples average011 8 samples average 100 16 samples average
1-0	Mode	R/W	0h	00 SW enabled, one-shot01 SW enabled, continuous 10 HW synchronized, one-shot11 HW synchronized, continuous

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.14 STEPDELAY9 Register (offset = 90h) [reset = 0h]

Step Delay Register 9

STEPDELAY9 is shown in Figure 11-18 and described in Table 11-19.

Figure 11-18. STEPDELAY9 Register

31	30	29	28	27	26	25	24
SampleDelay							
R/W-0h							

23	22	21	20	19	18	17	16
Reserved						OpenDelay	
-0h						R/W-0h	

15	14	13	12	11	10	9	8
OpenDelay							
R/W-0h							

7	6	5	4	3	2	1	0
OpenDelay							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-19. STEPDELAY9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SampleDelay	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SOC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	Reserved		0h	
17-0	OpenDelay	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.15 STEPCONFIG10 Register (offset = 94h) [reset = 0h]

Step Configuration 10

STEPCONFIG10 is shown in Figure 11-19 and described in Table 11-20.

Figure 11-19. STEPCONFIG10 Register

31	30	29	28	27	26	25	24
Reserved				Range_check	FIFO_select	Diff_CNTRL	SEL_RFM_SWC_1_0
-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
SEL_RFM_SWC_1_0	SEL_INP_SWC_3_0				SEL_INM_SWM3_0		
R/W-0h	R/W-0h				R/W-0h		

15	14	13	12	11	10	9	8
SEL_INM_SWM3_0	SEL_RFP_SWC_2_0			WPNSW_SWC	YPNSW_SWC	XNPSW_SWC	YNNSW_SWC
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
YPPSW_SWC	XNNSW_SWC	XPPSW_SWC	Averaging			Mode	
R/W-0h	R/W-0h	R/W-0h	R/W-0h			R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-20. STEPCONFIG10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved		0h	
27	Range_check	R/W	0h	0 disable out of range check 1 - Compare adc data with range check register
26	FIFO_select	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 01 -> FIFO 1
25	Diff_CNTRL	R/W	0h	Differential Control Pin
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins SW configuration
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins SW configuration
18-15	SEL_INM_SWM3_0	R/W	0h	SEL_INP pins for neg differential
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins SW configuration
11	WPNSW_SWC	R/W	0h	WPNSW pin SW configuration
10	YPNSW_SWC	R/W	0h	YPNSW pin SW configuration
9	XNPSW_SWC	R/W	0h	XNPSW pin SW configuration
8	YNNSW_SWC	R/W	0h	YNNSW pin SW configuration
7	YPPSW_SWC	R/W	0h	YPPSW pin SW configuration
6	XNNSW_SWC	R/W	0h	XNNSW pin SW configuration
5	XPPSW_SWC	R/W	0h	XPPSW pin SW configuration
4-2	Averaging	R/W	0h	Number of samplings to average: 000 -> no average001 2 samples average010 4 samples average011 8 samples average 100 16 samples average
1-0	Mode	R/W	0h	00 SW enabled, one-shot01 SW enabled, continuous 10 HW synchronized, one-shot11 HW synchronized, continuous

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.16 STEPDELAY10 Register (offset = 98h) [reset = 0h]

Step Delay Register 10

STEPDELAY10 is shown in Figure 11-20 and described in Table 11-21.

Figure 11-20. STEPDELAY10 Register

31	30	29	28	27	26	25	24
SampleDelay							
R/W-0h							

23	22	21	20	19	18	17	16
Reserved						OpenDelay	
-0h						R/W-0h	

15	14	13	12	11	10	9	8
OpenDelay							
R/W-0h							

7	6	5	4	3	2	1	0
OpenDelay							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-21. STEPDELAY10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SampleDelay	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SOC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	Reserved		0h	
17-0	OpenDelay	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.17 STEPCONFIG11 Register (offset = 9Ch) [reset = 0h]

Step Configuration 11

STEPCONFIG11 is shown in Figure 11-21 and described in Table 11-22.

Figure 11-21. STEPCONFIG11 Register

31	30	29	28	27	26	25	24
Reserved				Range_check	FIFO_select	Diff_CNTRL	SEL_RFM_SWC_1_0
-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
SEL_RFM_SWC_1_0	SEL_INP_SWC_3_0				SEL_INM_SWM3_0		
R/W-0h	R/W-0h				R/W-0h		

15	14	13	12	11	10	9	8
SEL_INM_SWM3_0	SEL_RFP_SWC_2_0			WPNSW_SWC	YPNSW_SWC	XNPSW_SWC	YNNSW_SWC
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
YPPSW_SWC	XNNSW_SWC	XPPSW_SWC	Averaging			Mode	
R/W-0h	R/W-0h	R/W-0h	R/W-0h			R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-22. STEPCONFIG11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved		0h	
27	Range_check	R/W	0h	0 disable out of range check 1 - Compare adc data with range check register
26	FIFO_select	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 01 -> FIFO 1
25	Diff_CNTRL	R/W	0h	Differential Control Pin
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins SW configuration
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins SW configuration
18-15	SEL_INM_SWM3_0	R/W	0h	SEL_INP pins for neg differential
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins SW configuration
11	WPNSW_SWC	R/W	0h	WPNSW pin SW configuration
10	YPNSW_SWC	R/W	0h	YPNSW pin SW configuration
9	XNPSW_SWC	R/W	0h	XNPSW pin SW configuration
8	YNNSW_SWC	R/W	0h	YNNSW pin SW configuration
7	YPPSW_SWC	R/W	0h	YPPSW pin SW configuration
6	XNNSW_SWC	R/W	0h	XNNSW pin SW configuration
5	XPPSW_SWC	R/W	0h	XPPSW pin SW configuration
4-2	Averaging	R/W	0h	Number of samplings to average: 000 -> no average001 2 samples average010 4 samples average011 8 samples average 100 16 samples average
1-0	Mode	R/W	0h	00 SW enabled, one-shot01 SW enabled, continuous 10 HW synchronized, one-shot11 HW synchronized, continuous

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.18 IRQSTATUS Register (offset = 10h) [reset = 0h]

IRQ status (masked)

IRQSTATUS is shown in Figure 12-55 and described in Table 12-42.

Figure 11-22. IRQSTATUS Register

31	30	29	28	27	26	25	24
Reserved							
-0h							

23	22	21	20	19	18	17	16
Reserved							
-0h							

15	14	13	12	11	10	9	8
Reserved					HW_Pen_Event_synchronous	Pen_Up_event	Out_of_Range
-0h					R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
FIFO1_Underflow	FIFO1_Overrun	FIFO1_Threshold	FIFO0_Underflow	FIFO0_Overrun	FIFO0_Threshold	End_of_Sequence	HW_Pen_Event_asynchronous
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 11-23. IRQSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	Reserved		0h	
10	HW_Pen_Event_synchronous	R/W	0h	Write 0: No action Read 0: No (enabled) event pending Read 1: Event pending Write 1: Clear (raw) event
9	Pen_Up_event	R/W	0h	Write 0: No action Read 0: No (enabled) event pending Read 1: Event pending Write 1: Clear (raw) event
8	Out_of_Range	R/W	0h	Write 0: No action Read 0: No (enabled) event pending Read 1: Event pending Write 1: Clear (raw) event
7	FIFO1_Underflow	R/W	0h	Write 0: No action Read 0: No (enabled) event pending Read 1: Event pending Write 1: Clear (raw) event
6	FIFO1_Overrun	R/W	0h	Write 0: No action Read 0: No (enabled) event pending Read 1: Event pending Write 1: Clear (raw) event
5	FIFO1_Threshold	R/W	0h	Write 0: No action Read 0: No (enabled) event pending Read 1: Event pending Write 1: Clear (raw) event
4	FIFO0_Underflow	R/W	0h	Write 0: No action Read 0: No (enabled) event pending Read 1: Event pending Write 1: Clear (raw) event
3	FIFO0_Overrun	R/W	0h	Write 0: No action Read 0: No (enabled) event pending Read 1: Event pending Write 1: Clear (raw) event

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 11-23. IRQSTATUS Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	FIFO0_Threshold	R/W	0h	Write 0: No action Read 0: No (enabled) event pending Read 1: Event pending Write 1: Clear (raw) event
1	End_of_Sequence	R/W	0h	Write 0: No action Read 0: No (enabled) event pending Read 1: Event pending Write 1: Clear (raw) event
0	HW_Pen_Event_asynchronous	R/W	0h	Write 0: No action Read 0: No (enabled) event pending Read 1: Event pending Write 1: Clear (raw) event

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.19 STEPDELAY11 Register (offset = A0h) [reset = 0h]

Step Delay Register 11

STEPDELAY11 is shown in Figure 11-23 and described in Table 11-24.

Figure 11-23. STEPDELAY11 Register

31	30	29	28	27	26	25	24
SampleDelay							
R/W-0h							

23	22	21	20	19	18	17	16
Reserved						OpenDelay	
-0h						R/W-0h	

15	14	13	12	11	10	9	8
OpenDelay							
R/W-0h							

7	6	5	4	3	2	1	0
OpenDelay							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-24. STEPDELAY11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SampleDelay	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SOC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	Reserved		0h	
17-0	OpenDelay	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.20 STEPCONFIG12 Register (offset = A4h) [reset = 0h]

Step Configuration 12

STEPCONFIG12 is shown in Figure 11-24 and described in Table 11-25.

Figure 11-24. STEPCONFIG12 Register

31	30	29	28	27	26	25	24
Reserved				Range_check	FIFO_select	Diff_CNTRL	SEL_RFM_SWC_1_0
-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
SEL_RFM_SWC_1_0	SEL_INP_SWC_3_0				SEL_INM_SWM3_0		
R/W-0h	R/W-0h				R/W-0h		

15	14	13	12	11	10	9	8
SEL_INM_SWM3_0	SEL_RFP_SWC_2_0			WPNSW_SWC	YPNSW_SWC	XNPSW_SWC	YNNSW_SWC
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
YPPSW_SWC	XNNSW_SWC	XPPSW_SWC	Averaging			Mode	
R/W-0h	R/W-0h	R/W-0h	R/W-0h			R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-25. STEPCONFIG12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved		0h	
27	Range_check	R/W	0h	0 disable out of range check 1 - Compare adc data with range check register
26	FIFO_select	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 01 -> FIFO 1
25	Diff_CNTRL	R/W	0h	Differential Control Pin
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins SW configuration
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins SW configuration
18-15	SEL_INM_SWM3_0	R/W	0h	SEL_INP pins for neg differential
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins SW configuration
11	WPNSW_SWC	R/W	0h	WPNSW pin SW configuration
10	YPNSW_SWC	R/W	0h	YPNSW pin SW configuration
9	XNPSW_SWC	R/W	0h	XNPSW pin SW configuration
8	YNNSW_SWC	R/W	0h	YNNSW pin SW configuration
7	YPPSW_SWC	R/W	0h	YPPSW pin SW configuration
6	XNNSW_SWC	R/W	0h	XNNSW pin SW configuration
5	XPPSW_SWC	R/W	0h	XPPSW pin SW configuration
4-2	Averaging	R/W	0h	Number of samplings to average: 000 -> no average001 2 samples average010 4 samples average011 8 samples average 100 16 samples average
1-0	Mode	R/W	0h	00 SW enabled, one-shot01 SW enabled, continuous 10 HW synchronized, one-shot11 HW synchronized, continuous

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.21 STEPDELAY12 Register (offset = A8h) [reset = 0h]

Step Delay Register 12

STEPDELAY12 is shown in Figure 11-25 and described in Table 11-26.

Figure 11-25. STEPDELAY12 Register

31	30	29	28	27	26	25	24
SampleDelay							
R/W-0h							

23	22	21	20	19	18	17	16
Reserved						OpenDelay	
-0h						R/W-0h	

15	14	13	12	11	10	9	8
OpenDelay							
R/W-0h							

7	6	5	4	3	2	1	0
OpenDelay							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-26. STEPDELAY12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SampleDelay	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SOC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	Reserved		0h	
17-0	OpenDelay	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.22 STEPCONFIG13 Register (offset = ACh) [reset = 0h]

Step Configuration 13

STEPCONFIG13 is shown in Figure 11-26 and described in Table 11-27.

Figure 11-26. STEPCONFIG13 Register

31	30	29	28	27	26	25	24
Reserved				Range_check	FIFO_select	Diff_CNTRL	SEL_RFM_SWC_1_0
-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
SEL_RFM_SWC_1_0	SEL_INP_SWC_3_0				SEL_INM_SWM3_0		
R/W-0h	R/W-0h				R/W-0h		

15	14	13	12	11	10	9	8
SEL_INM_SWM3_0	SEL_RFP_SWC_2_0			WPNSW_SWC	YPNSW_SWC	XNPSW_SWC	YNNSW_SWC
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
YPPSW_SWC	XNNSW_SWC	XPPSW_SWC	Averaging			Mode	
R/W-0h	R/W-0h	R/W-0h	R/W-0h			R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-27. STEPCONFIG13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved		0h	
27	Range_check	R/W	0h	0 disable out of range check 1 - Compare adc data with range check register
26	FIFO_select	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 01 -> FIFO 1
25	Diff_CNTRL	R/W	0h	Differential Control Pin
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins SW configuration
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins SW configuration
18-15	SEL_INM_SWM3_0	R/W	0h	SEL_INP pins for neg differential
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins SW configuration
11	WPNSW_SWC	R/W	0h	WPNSW pin SW configuration
10	YPNSW_SWC	R/W	0h	YPNSW pin SW configuration
9	XNPSW_SWC	R/W	0h	XNPSW pin SW configuration
8	YNNSW_SWC	R/W	0h	YNNSW pin SW configuration
7	YPPSW_SWC	R/W	0h	YPPSW pin SW configuration
6	XNNSW_SWC	R/W	0h	XNNSW pin SW configuration
5	XPPSW_SWC	R/W	0h	XPPSW pin SW configuration
4-2	Averaging	R/W	0h	Number of samplings to average: 000 -> no average001 2 samples average010 4 samples average011 8 samples average 100 16 samples average
1-0	Mode	R/W	0h	00 SW enabled, one-shot01 SW enabled, continuous 10 HW synchronized, one-shot11 HW synchronized, continuous

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.23 STEPDELAY13 Register (offset = B0h) [reset = 0h]

Step Delay Register 13

STEPDELAY13 is shown in Figure 11-27 and described in Table 11-28.

Figure 11-27. STEPDELAY13 Register

31	30	29	28	27	26	25	24
SampleDelay							
R/W-0h							

23	22	21	20	19	18	17	16
Reserved						OpenDelay	
-0h						R/W-0h	

15	14	13	12	11	10	9	8
OpenDelay							
R/W-0h							

7	6	5	4	3	2	1	0
OpenDelay							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-28. STEPDELAY13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SampleDelay	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SOC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	Reserved		0h	
17-0	OpenDelay	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.24 STEPCONFIG14 Register (offset = B4h) [reset = 0h]

Step Configuration 14

STEPCONFIG14 is shown in Figure 11-28 and described in Table 11-29.

Figure 11-28. STEPCONFIG14 Register

31	30	29	28	27	26	25	24
Reserved				Range_check	FIFO_select	Diff_CNTRL	SEL_RFM_SWC_1_0
-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
SEL_RFM_SWC_1_0	SEL_INP_SWC_3_0				SEL_INM_SWM3_0		
R/W-0h	R/W-0h				R/W-0h		

15	14	13	12	11	10	9	8
SEL_INM_SWM3_0	SEL_RFP_SWC_2_0			WPNSW_SWC	YPNSW_SWC	XNPSW_SWC	YNNSW_SWC
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
YPPSW_SWC	XNNSW_SWC	XPPSW_SWC	Averaging			Mode	
R/W-0h	R/W-0h	R/W-0h	R/W-0h			R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-29. STEPCONFIG14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved		0h	
27	Range_check	R/W	0h	0 disable out of range check 1 - Compare adc data with range check register
26	FIFO_select	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 01 -> FIFO 1
25	Diff_CNTRL	R/W	0h	Differential Control Pin
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins SW configuration
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins SW configuration
18-15	SEL_INM_SWM3_0	R/W	0h	SEL_INP pins for neg differential
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins SW configuration
11	WPNSW_SWC	R/W	0h	WPNSW pin SW configuration
10	YPNSW_SWC	R/W	0h	YPNSW pin SW configuration
9	XNPSW_SWC	R/W	0h	XNPSW pin SW configuration
8	YNNSW_SWC	R/W	0h	YNNSW pin SW configuration
7	YPPSW_SWC	R/W	0h	YPPSW pin SW configuration
6	XNNSW_SWC	R/W	0h	XNNSW pin SW configuration
5	XPPSW_SWC	R/W	0h	XPPSW pin SW configuration
4-2	Averaging	R/W	0h	Number of samplings to average: 000 -> no average001 2 samples average010 4 samples average011 8 samples average 100 16 samples average
1-0	Mode	R/W	0h	00 SW enabled, one-shot01 SW enabled, continuous 10 HW synchronized, one-shot11 HW synchronized, continuous

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.25 STEPDELAY14 Register (offset = B8h) [reset = 0h]

Step Delay Register 14

STEPDELAY14 is shown in Figure 11-29 and described in Table 11-30.

Figure 11-29. STEPDELAY14 Register

31	30	29	28	27	26	25	24
SampleDelay							
R/W-0h							

23	22	21	20	19	18	17	16
Reserved						OpenDelay	
-0h						R/W-0h	

15	14	13	12	11	10	9	8
OpenDelay							
R/W-0h							

7	6	5	4	3	2	1	0
OpenDelay							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-30. STEPDELAY14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SampleDelay	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SOC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	Reserved		0h	
17-0	OpenDelay	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.26 STEPCONFIG15 Register (offset = BCh) [reset = 0h]

Step Configuration 15

STEPCONFIG15 is shown in Figure 11-30 and described in Table 11-31.

Figure 11-30. STEPCONFIG15 Register

31	30	29	28	27	26	25	24
Reserved				Range_check	FIFO_select	Diff_CNTRL	SEL_RFM_SWC_1_0
-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
SEL_RFM_SWC_1_0	SEL_INP_SWC_3_0				SEL_INM_SWM3_0		
R/W-0h	R/W-0h				R/W-0h		

15	14	13	12	11	10	9	8
SEL_INM_SWM3_0	SEL_RFP_SWC_2_0			WPNSW_SWC	YPNSW_SWC	XNPSW_SWC	YNNSW_SWC
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
YPPSW_SWC	XNNSW_SWC	XPPSW_SWC	Averaging			Mode	
R/W-0h	R/W-0h	R/W-0h	R/W-0h			R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-31. STEPCONFIG15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved		0h	
27	Range_check	R/W	0h	0 disable out of range check 1 - Compare adc data with range check register
26	FIFO_select	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 01 -> FIFO 1
25	Diff_CNTRL	R/W	0h	Differential Control Pin
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins SW configuration
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins SW configuration
18-15	SEL_INM_SWM3_0	R/W	0h	SEL_INP pins for neg differential
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins SW configuration
11	WPNSW_SWC	R/W	0h	WPNSW pin SW configuration
10	YPNSW_SWC	R/W	0h	YPNSW pin SW configuration
9	XNPSW_SWC	R/W	0h	XNPSW pin SW configuration
8	YNNSW_SWC	R/W	0h	YNNSW pin SW configuration
7	YPPSW_SWC	R/W	0h	YPPSW pin SW configuration
6	XNNSW_SWC	R/W	0h	XNNSW pin SW configuration
5	XPPSW_SWC	R/W	0h	XPPSW pin SW configuration
4-2	Averaging	R/W	0h	Number of samplings to average: 000 -> no average001 2 samples average010 4 samples average011 8 samples average 100 16 samples average
1-0	Mode	R/W	0h	00 SW enabled, one-shot01 SW enabled, continuous 10 HW synchronized, one-shot11 HW synchronized, continuous

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.27 STEPDELAY15 Register (offset = C0h) [reset = 0h]

Step Delay Register 15

STEPDELAY15 is shown in Figure 11-31 and described in Table 11-32.

Figure 11-31. STEPDELAY15 Register

31	30	29	28	27	26	25	24
SampleDelay							
R/W-0h							

23	22	21	20	19	18	17	16
Reserved						OpenDelay	
-0h						R/W-0h	

15	14	13	12	11	10	9	8
OpenDelay							
R/W-0h							

7	6	5	4	3	2	1	0
OpenDelay							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-32. STEPDELAY15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SampleDelay	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SOC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	Reserved		0h	
17-0	OpenDelay	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.28 STEPCONFIG16 Register (offset = C4h) [reset = 0h]

Step Configuration 16

STEPCONFIG16 is shown in Figure 11-32 and described in Table 11-33.

Figure 11-32. STEPCONFIG16 Register

31	30	29	28	27	26	25	24
Reserved				Range_check	FIFO_select	Diff_CNTRL	SEL_RFM_SWC_1_0
-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
SEL_RFM_SWC_1_0	SEL_INP_SWC_3_0				SEL_INM_SWM3_0		
R/W-0h	R/W-0h				R/W-0h		

15	14	13	12	11	10	9	8
SEL_INM_SWM3_0	SEL_RFP_SWC_2_0			WPNSW_SWC	YPNSW_SWC	XNPSW_SWC	YNNSW_SWC
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
YPPSW_SWC	XNNSW_SWC	XPPSW_SWC	Averaging			Mode	
R/W-0h	R/W-0h	R/W-0h	R/W-0h			R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-33. STEPCONFIG16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved		0h	
27	Range_check	R/W	0h	0 disable out of range check 1 - Compare adc data with range check register
26	FIFO_select	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 01 -> FIFO 1
25	Diff_CNTRL	R/W	0h	Differential Control Pin
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins SW configuration
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins SW configuration
18-15	SEL_INM_SWM3_0	R/W	0h	SEL_INP pins for neg differential
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins SW configuration
11	WPNSW_SWC	R/W	0h	WPNSW pin SW configuration
10	YPNSW_SWC	R/W	0h	YPNSW pin SW configuration
9	XNPSW_SWC	R/W	0h	XNPSW pin SW configuration
8	YNNSW_SWC	R/W	0h	YNNSW pin SW configuration
7	YPPSW_SWC	R/W	0h	YPPSW pin SW configuration
6	XNNSW_SWC	R/W	0h	XNNSW pin SW configuration
5	XPPSW_SWC	R/W	0h	XPPSW pin SW configuration
4-2	Averaging	R/W	0h	Number of samplings to average: 000 -> no average001 2 samples average010 4 samples average011 8 samples average 100 16 samples average
1-0	Mode	R/W	0h	00 SW enabled, one-shot01 SW enabled, continuous 10 HW synchronized, one-shot11 HW synchronized, continuous

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.29 IRQENABLE_SET Register (offset = 14h) [reset = 0h]

IRQ enable set bits

IRQENABLE_SET is shown in Figure 12-56 and described in Table 12-43.

Figure 11-33. IRQENABLE_SET Register

31	30	29	28	27	26	25	24
Reserved							
-0h							

23	22	21	20	19	18	17	16
Reserved							
-0h							

15	14	13	12	11	10	9	8
Reserved					HW_Pen_Event_synchronous	Pen_Up_event	Out_of_Range
-0h					R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
FIFO1_Underflow	FIFO1_Overrun	FIFO1_Threshold	FIFO0_Underflow	FIFO0_Overrun	FIFO0_Threshold	End_of_Sequence	HW_Pen_Event_asynchronous
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 11-34. IRQENABLE_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	Reserved		0h	
10	HW_Pen_Event_synchronous	R/W	0h	Write 0: No action Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled Write 1: Enable interrupt
9	Pen_Up_event	R/W	0h	Write 0: No action Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled Write 1: Enable interrupt
8	Out_of_Range	R/W	0h	Write 0: No action Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled Write 1: Enable interrupt
7	FIFO1_Underflow	R/W	0h	Write 0: No action Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled Write 1: Enable interrupt
6	FIFO1_Overrun	R/W	0h	Write 0: No action Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled Write 1: Enable interrupt
5	FIFO1_Threshold	R/W	0h	Write 0: No action Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled Write 1: Enable interrupt
4	FIFO0_Underflow	R/W	0h	Write 0: No action Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled Write 1: Enable interrupt
3	FIFO0_Overrun	R/W	0h	Write 0: No action Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled Write 1: Enable interrupt

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 11-34. IRQENABLE_SET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	FIFO0_Threshold	R/W	0h	Write 0: No action Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled Write 1: Enable interrupt
1	End_of_Sequence	R/W	0h	Write 0: No action Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled Write 1: Enable interrupt
0	HW_Pen_Event_asynchronous	R/W	0h	Write 0: No action Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled Write 1: Enable interrupt

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.30 STEPDELAY16 Register (offset = C8h) [reset = 0h]

Step Delay Register 16

STEPDELAY16 is shown in Figure 11-34 and described in Table 11-35.

Figure 11-34. STEPDELAY16 Register

31	30	29	28	27	26	25	24
SampleDelay							
R/W-0h							

23	22	21	20	19	18	17	16
Reserved						OpenDelay	
-0h						R/W-0h	

15	14	13	12	11	10	9	8
OpenDelay							
R/W-0h							

7	6	5	4	3	2	1	0
OpenDelay							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-35. STEPDELAY16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SampleDelay	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SOC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	Reserved		0h	
17-0	OpenDelay	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.31 FIFO0COUNT Register (offset = CCh) [reset = 0h]

FIFO0 word count@TSC_ADC_SS_FIFO0 Word Count Register

FIFO0COUNT is shown in Figure 11-35 and described in Table 11-36.

Figure 11-35. FIFO0COUNT Register

31	30	29	28	27	26	25	24
Reserved							
R-0h							

23	22	21	20	19	18	17	16
Reserved							
R-0h							

15	14	13	12	11	10	9	8
Reserved							
R-0h							

7	6	5	4	3	2	1	0
Reserved	Words_in_FIFO0						
R-0h	R-0h						

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-36. FIFO0COUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	Reserved	R	0h	RESERVED
6-0	Words_in_FIFO0	R	0h	Number of words currently in the FIFO0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.32 FIFO0THRESHOLD Register (offset = D0h) [reset = 0h]

FIFO0 Threshold trigger@TSC_ADC_SS_FIFO0 Threshold Level Register

FIFO0THRESHOLD is shown in Figure 11-36 and described in Table 11-37.

Figure 11-36. FIFO0THRESHOLD Register

31	30	29	28	27	26	25	24
Reserved							
R-0h							

23	22	21	20	19	18	17	16
Reserved							
R-0h							

15	14	13	12	11	10	9	8
Reserved							
R-0h							

7	6	5	4	3	2	1	0
Reserved		FIFO0_threshold_Level					
R-0h		R/W-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-37. FIFO0THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	Reserved	R	0h	
5-0	FIFO0_threshold_Level	R/W	0h	Program the desired FIFO0 data sample level to reach before generating interrupt to CPU (program to value minus 1)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.33 DMA0REQ Register (offset = D4h) [reset = 0h]

FIFO0 DMA req0 trigger@TSC_ADC_SS_FIFO0 DMA REQUEST Register

DMA0REQ is shown in Figure 11-37 and described in Table 11-38.

Figure 11-37. DMA0REQ Register

31	30	29	28	27	26	25	24
Reserved							
R-0h							

23	22	21	20	19	18	17	16
Reserved							
R-0h							

15	14	13	12	11	10	9	8
Reserved							
R-0h							

7	6	5	4	3	2	1	0
Reserved		DMA_Request_Level					
R-0h		R/W-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-38. DMA0REQ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	Reserved	R	0h	RESERVED
5-0	DMA_Request_Level	R/W	0h	Number of words in FIFO0 before generating a DMA request (program to value minus 1)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.34 FIFO1COUNT Register (offset = D8h) [reset = 0h]

FIFO1 word count@TSC_ADC_SS_FIFO1 Word Count Register

FIFO1COUNT is shown in Figure 11-38 and described in Table 11-39.

Figure 11-38. FIFO1COUNT Register

31	30	29	28	27	26	25	24
Reserved							
R-0h							

23	22	21	20	19	18	17	16
Reserved							
R-0h							

15	14	13	12	11	10	9	8
Reserved							
R-0h							

7	6	5	4	3	2	1	0
Reserved	Words_in_FIFO0						
R-0h	R-0h						

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-39. FIFO1COUNT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-7	Reserved	R	0h	RESERVED
6-0	Words_in_FIFO0	R	0h	Number of words currently in the FIFO0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.35 FIFO1THRESHOLD Register (offset = DCh) [reset = 0h]

FIFO1 Threshold trigger@TSC_ADC_SS_FIFO1 Threshold Level Register

FIFO1THRESHOLD is shown in Figure 11-39 and described in Table 11-40.

Figure 11-39. FIFO1THRESHOLD Register

31	30	29	28	27	26	25	24
Reserved							
R-0h							

23	22	21	20	19	18	17	16
Reserved							
R-0h							

15	14	13	12	11	10	9	8
Reserved							
R-0h							

7	6	5	4	3	2	1	0
Reserved		FIFO0_threshold_Level					
R-0h		R/W-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-40. FIFO1THRESHOLD Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	Reserved	R	0h	
5-0	FIFO0_threshold_Level	R/W	0h	Program the desired FIFO0 data sample level to reach before generating interrupt to CPU (program to value minus 1)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.36 DMA1REQ Register (offset = E0h) [reset = 0h]

FIFO1 DMA req1 trigger@TSC_ADC_SS_FIFO1 DMA Request Register

DMA1REQ is shown in Figure 11-40 and described in Table 11-41.

Figure 11-40. DMA1REQ Register

31	30	29	28	27	26	25	24
Reserved							
R-0h							

23	22	21	20	19	18	17	16
Reserved							
R-0h							

15	14	13	12	11	10	9	8
Reserved							
R-0h							

7	6	5	4	3	2	1	0
Reserved		DMA_Request_Level					
R-0h		R/W-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-41. DMA1REQ Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	Reserved	R	0h	RESERVED
5-0	DMA_Request_Level	R/W	0h	Number of words in FIFO0 before generating a DMA request (program to value minus 1)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.37 FIFO0DATA Register (offset = E4h) [reset = 0h]

ADC_ FIFO0 _READ Data @TSC_ADC_SS_FIFO0 READ Register

FIFO0DATA is shown in Figure 11-41 and described in Table 11-42.

Figure 11-41. FIFO0DATA Register

31	30	29	28	27	26	25	24
Reserved							
R-0h							

23	22	21	20	19	18	17	16
Reserved				ADCCHNLID			
R-0h				R-0h			

15	14	13	12	11	10	9	8
Reserved				ADCDATA			
				R-0h			

7	6	5	4	3	2	1	0
ADCDATA							
R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-42. FIFO0DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	Reserved	R	0h	RESERVED.
19-16	ADCCHNLID	R	0h	Optional ID tag of channel that captured the data. If tag option is disabled, these bits will be 0.
15-12	Reserved			
11-0	ADCDATA	R	0h	12 bit sampled ADC converted data value stored in FIFO 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.38 FIFO1DATA Register (offset = E8h) [reset = 0h]

ADC FIFO1_READ Data@TSC_ADC_SS_FIFO1 READ Register

FIFO1DATA is shown in Figure 11-42 and described in Table 11-43.

Figure 11-42. FIFO1DATA Register

31	30	29	28	27	26	25	24
Reserved							
R-0h							

23	22	21	20	19	18	17	16
Reserved				ADCCHNLID			
R-0h				R-0h			

15	14	13	12	11	10	9	8
Reserved				ADCDATA			
R-0h				R-0h			

7	6	5	4	3	2	1	0
ADCDATA							
R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-43. FIFO1DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	Reserved	R	0h	RESERVED
19-16	ADCCHNLID	R	0h	Optional ID tag of channel that captured the data. If tag option is disabled, these bits will be 0.
15-12	Reserved	R	0h	RESERVED
11-0	ADCDATA	R	0h	12 bit sampled ADC converted data value stored in FIFO 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.39 IRQENABLE_CLR Register (offset = 18h) [reset = 0h]

IRQ enable clear bits

IRQENABLE_CLR is shown in Figure 11-43 and described in Table 11-44.

Figure 11-43. IRQENABLE_CLR Register

31	30	29	28	27	26	25	24
Reserved							
-0h							

23	22	21	20	19	18	17	16
Reserved							
-0h							

15	14	13	12	11	10	9	8
Reserved					HW_Pen_Event_synchronous	Pen_Up_Event	Out_of_Range
-0h					R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
FIFO1_Underflow	FIFO1_Overrun	FIFO1_Threshold	FIFO0_Underflow	FIFO0_Overrun	FIFO0_Threshold	End_of_Sequence	HW_Pen_Event_asynchronous
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 11-44. IRQENABLE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-11	Reserved		0h	
10	HW_Pen_Event_synchronous	R/W	0h	Write 0: No action Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled Write 1: Disable interrupt
9	Pen_Up_Event	R/W	0h	Write 0: No action Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled Write 1: Disable interrupt
8	Out_of_Range	R/W	0h	Write 0: No action Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled Write 1: Disable interrupt
7	FIFO1_Underflow	R/W	0h	Write 0: No action Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled Write 1: Disable interrupt
6	FIFO1_Overrun	R/W	0h	Write 0: No action Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled Write 1: Disable interrupt
5	FIFO1_Threshold	R/W	0h	Write 0: No action Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled Write 1: Disable interrupt
4	FIFO0_Underflow	R/W	0h	Write 0: No action Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled Write 1: Disable interrupt
3	FIFO0_Overrun	R/W	0h	Write 0: No action Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled Write 1: Disable interrupt

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 11-44. IRQENABLE_CLR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2	FIFO0_Threshold	R/W	0h	Write 0: No action Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled Write 1: Disable interrupt
1	End_of_Sequence	R/W	0h	Write 0: No action Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled Write 1: Disable interrupt
0	HW_Pen_Event_asynchronous	R/W	0h	Write 0: No action Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled Write 1: Disable interrupt

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.40 IRQWAKEUP Register (offset = 1Ch) [reset = 0h]

IRQ wakeup enable

IRQWAKEUP is shown in Figure 11-44 and described in Table 11-45.

Figure 11-44. IRQWAKEUP Register

31	30	29	28	27	26	25	24
Reserved							
-0h							

23	22	21	20	19	18	17	16
Reserved							
-0h							

15	14	13	12	11	10	9	8
Reserved							
-0h							

7	6	5	4	3	2	1	0
Reserved							WAKEEN0
-0h							R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-45. IRQWAKEUP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved		0h	
0	WAKEEN0	R/W	0h	Wakeup generation for HW Pen event 0: Wakeup disabled 1: Wakeup enabled

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.41 DMAENABLE_SET Register (offset = 20h) [reset = 0h]

Per-Line DMA set

DMAENABLE_SET is shown in Figure 11-45 and described in Table 11-46.

Figure 11-45. DMAENABLE_SET Register

31	30	29	28	27	26	25	24
Reserved							
-0h							

23	22	21	20	19	18	17	16
Reserved							
-0h							

15	14	13	12	11	10	9	8
Reserved							
-0h							

7	6	5	4	3	2	1	0
Reserved						Enable_1	Enable_0
-0h						R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-46. DMAENABLE_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	Reserved		0h	
1	Enable_1	R/W	0h	Enable DMA request FIFO 1 Write 0 : No action Read 0 : DMA line disabled Read 1: DMA line enabled Write 1: Enable DMA line
0	Enable_0	R/W	0h	Enable DMA request FIFO 0 Write 0 : No action Read 0 : DMA line disabled Read 1: DMA line enabled Write 1: Enable DMA line

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.42 DMAENABLE_CLR Register (offset = 24h) [reset = 0h]

Per-Line DMA clr

DMAENABLE_CLR is shown in Figure 11-46 and described in Table 11-47.

Figure 11-46. DMAENABLE_CLR Register

31	30	29	28	27	26	25	24
Reserved							
-0h							

23	22	21	20	19	18	17	16
Reserved							
-0h							

15	14	13	12	11	10	9	8
Reserved							
-0h							

7	6	5	4	3	2	1	0
Reserved						Enable_1	Enable_0
-0h						R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-47. DMAENABLE_CLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	Reserved		0h	
1	Enable_1	R/W	0h	Disable DMA request FIFO 1 Write 0 : No action Read 0 : DMA line disabled Read 1: DMA line enabled Write 1: Disable DMA line
0	Enable_0	R/W	0h	Disable DMA request FIFO 0 Write 0 : No action Read 0 : DMA line disabled Read 1: DMA line enabled Write 1: Disable DMA line

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.43 SYSCONFIG Register (offset = 4h) [reset = 8h]

SysConfig Register

SYSCONFIG is shown in Figure 16-2 and described in Table 16-5.

Figure 11-47. SYSCONFIG Register

31	30	29	28	27	26	25	24
Reserved							
-0h							

23	22	21	20	19	18	17	16
Reserved							
-0h							

15	14	13	12	11	10	9	8
Reserved							
-0h							

7	6	5	4	3	2	1	0
Reserved				IdleMode		Reserved	
-0h				R/W-2h		-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-48. SYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	Reserved		0h	
3-2	IdleMode	R/W	2h	00 -> Force Idle (always acknowledges) 01 -> No Idle Mode (never acknowledges) 10 -> Smart-Idle Mode 11 -> Smart Idle with Wakeup
1-0	Reserved		0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.44 CTRL Register (offset = 28h) [reset = 0h]

@TSC_ADC_SS Control Register

CTRL is shown in Figure 12-41 and described in Table 12-28.

Figure 11-48. CTRL Register

31	30	29	28	27	26	25	24
Reserved							
R-0h							

23	22	21	20	19	18	17	16
Reserved							
R-0h							

15	14	13	12	11	10	9	8
Reserved						HW_preempt	HW_event_mapping
R-0h						R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
Touch_Screen_Enable	AFE_Pen_Ctrl		Power_Down	ADC_Bias_Select	StepConfig_WriteProtect_n_active_low	Step_ID_tag	Enable
R/W-0h	R/W-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 11-49. CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	Reserved	R	0h	
9	HW_preempt	R/W	0h	0->SW steps are not preempted by HW events 1-> SW steps are preempted by HW events
8	HW_event_mapping	R/W	0h	0->Map HW event to Pen touch irq (from AFE) 1->Map HW event to HW event input
7	Touch_Screen_Enable	R/W	0h	0 -> Touch Screen transistors disabled 1 -> TouchScreen transistors enabled
6-5	AFE_Pen_Ctrl	R/W	0h	These 2 bits are sent directly to the AFE Pen Ctrl inputs. Bit 6 controls the Wiper touch (5 wire modes)Bit 5 controls the X+ touch (4 wire modes)User also needs to make sure the ground path is connected properly for pen interrupt to occur (using the StepConfig registers)Refer to section 4 interrupts for more information
4	Power_Down	R/W	0h	ADC Power Down control 0: AFE is powered up (default) 1 : Write 1 to power down AFE (the tsc_adc_ss enable (bit 0) should also be set to off)
3	ADC_Bias_Select	R/W	0h	Select Internal or External Bias to AFE 0 -> Internal 1 -> External
2	StepConfig_WriteProtect_n_active_low	R/W	0h	0-> Step Configuration Registers are protected (not writable) 1-> Step Configuration are not protected (writable)
1	Step_ID_tag	R/W	0h	Writing 1 to this bit will store the Step ID number with the captured ADC data in the FIFO. 0 write zeros1 store the channel id tag
0	Enable	R/W	0h	TSC_ADC_SS module enable bit. After programming all the steps and configuration registers, write a 1 to this bit to turn on TSC_ADC_SS. Writing a 0 will disable the module (after the current conversion)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.45 ADCSTAT Register (offset = 2Ch) [reset = 10h]

General Status bits @TSC_ADC_SS_Sequencer_Status Register

ADCSTAT is shown in Figure 11-49 and described in Table 11-50.

Figure 11-49. ADCSTAT Register

31	30	29	28	27	26	25	24
Reserved							
R-0h							

23	22	21	20	19	18	17	16
Reserved							
R-0h							

15	14	13	12	11	10	9	8
Reserved							
R-0h							

7	6	5	4	3	2	1	0
PEN_IRQ1	PEN_IRQ0	FSM_BUSY	STEP_ID				
R-0h	R-0h	R-0h	R-10h				

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-50. ADCSTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	Reserved	R	0h	-
7	PEN_IRQ1	R	0h	PEN_IRQ[1] status
6	PEN_IRQ0	R	0h	PEN_IRQ[0] status
5	FSM_BUSY	R	0h	Status of OCP FSM and ADC FSM0 idle 1 - busy
4-0	STEP_ID	R	10h	Below are encoded values:10000 Idle10001 Charge00000 Step 100001 Step 200010 Step 300011 Step 400100 Step 500101 Step 600110 Step 700111 Step 801000 Step 901001 Step 1001010 Step 1101011 Step 1201100 Step 1301101 Step 1401110 Step 1501111 Step 16

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.46 ADCRANGE Register (offset = 30h) [reset = 0h]

High and Low Range Threshold@TSC_ADC_SS_Range_Check Register

ADCRANGE is shown in Figure 11-50 and described in Table 11-51.

Figure 11-50. ADCRANGE Register

31	30	29	28	27	26	25	24
Reserved				High_Range_Data			
R-0h				R/W-0h			

23	22	21	20	19	18	17	16
High_Range_Data							
R/W-0h							

15	14	13	12	11	10	9	8
High_Range_Data				Low_Range_Data			
R/W-0h				R/W-0h			

7	6	5	4	3	2	1	0
Low_Range_Data							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-51. ADCRANGE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved	R	0h	
27-12	High_Range_Data	R/W	0h	Sampled ADC data is compared to this value. If the sampled data is > value, then interrupt is generated.
11-0	Low_Range_Data	R/W	0h	Sampled ADC data is compared to this value. If the sampled data is and#60 value then interrupt is generated.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.47 ADC_CLKDIV Register (offset = 34h) [reset = 0h]

ADC clock divider register@TSC_ADC_SS_Clock_Divider Register

ADC_CLKDIV is shown in Figure 11-51 and described in Table 11-52.

Figure 11-51. ADC_CLKDIV Register

31	30	29	28	27	26	25	24
Reserved							
R-0h							

23	22	21	20	19	18	17	16
Reserved							
R-0h							

15	14	13	12	11	10	9	8
ADC_ClkDiv							
R/W-0h							

7	6	5	4	3	2	1	0
ADC_ClkDiv							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-52. ADC_CLKDIV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	Reserved	R	0h	
15-0	ADC_ClkDiv	R/W	0h	The input ADC clock will be divided by this value and sent to the AFE. Program to the value minus 1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.48 ADC_MISC Register (offset = 38h) [reset = 0h]

AFE misc debug@TSC_ADC_SS_MISC Register

ADC_MISC is shown in Figure 11-52 and described in Table 11-53.

Figure 11-52. ADC_MISC Register

31	30	29	28	27	26	25	24
Reserved							
R-0h							

23	22	21	20	19	18	17	16
Reserved							
R-0h							

15	14	13	12	11	10	9	8
Reserved							
R-0h							

7	6	5	4	3	2	1	0
AFE_Spare_Output				AFE_Spare_Input			
R-0h				R/W-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-53. ADC_MISC Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	Reserved	R	0h	RESERVED.
7-4	AFE_Spare_Output	R	0h	Connected to AFE Spare Output pins
3-0	AFE_Spare_Input	R/W	0h	Connected to AFE Spare Input pins

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.49 STEPENABLE Register (offset = 3Ch) [reset = 0h]

Step Enable

STEPENABLE is shown in Figure 11-53 and described in Table 11-54.

Figure 11-53. STEPENABLE Register

31	30	29	28	27	26	25	24
Reserved							
R-0h							

23	22	21	20	19	18	17	16
Reserved							STEP16
R-0h							R/W-0h

15	14	13	12	11	10	9	8
STEP15	STEP14	STEP13	STEP12	STEP11	STEP10	STEP9	STEP8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
STEP7	STEP6	STEP5	STEP4	STEP3	STEP2	STEP1	TS_Charge
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-54. STEPENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-17	Reserved	R	0h	RESERVED.
16	STEP16	R/W	0h	Enable step 16
15	STEP15	R/W	0h	Enable step 15
14	STEP14	R/W	0h	Enable step 14
13	STEP13	R/W	0h	Enable step 13
12	STEP12	R/W	0h	Enable step 12
11	STEP11	R/W	0h	Enable step 11
10	STEP10	R/W	0h	Enable step 10
9	STEP9	R/W	0h	Enable step 9
8	STEP8	R/W	0h	Enable step 8
7	STEP7	R/W	0h	Enable step 7
6	STEP6	R/W	0h	Enable step 6
5	STEP5	R/W	0h	Enable step 5
4	STEP4	R/W	0h	Enable step 4
3	STEP3	R/W	0h	Enable step 3
2	STEP2	R/W	0h	Enable step 2
1	STEP1	R/W	0h	Enable step 1
0	TS_Charge	R/W	0h	Enable TS Charge step

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.50 IDLECONFIG Register (offset = 40h) [reset = 0h]

Idle Step configuration@TSC_ADC_SS_IDLE_StepConfig Register

IDLECONFIG is shown in Figure 11-54 and described in Table 11-55.

Figure 11-54. IDLECONFIG Register

31	30	29	28	27	26	25	24
Reserved						Diff_CNTRL	SEL_RFM__SWC_1_0
-0h						R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
SEL_RFM__SWC_1_0	SEL_INP_SWC_3_0				SEL_INM_SWM3_0		
R/W-0h	R/W-0h				R/W-0h		

15	14	13	12	11	10	9	8
SEL_INM_SWM3_0	SEL_RFP__SWC_2_0			WPNSW_SWC	YPNSW_SWC	XNPSW_SWC	YNNSW_SWC
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
YPPSW__SWC	XNNSW__SWC	XPPSW_SWC	Reserved				
R/W-0h	R/W-0h	R/W-0h	-0h				

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 11-55. IDLECONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved		0h	
25	Diff_CNTRL	R/W	0h	Differential Control Pin 0 -> Single Ended 1 -> Differential Pair Enable
24-23	SEL_RFM__SWC_1_0	R/W	0h	SEL_RFM pins SW configuration 00 VSSA01- XNUR10 YNLR11 ADCREFM
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins SW configuration 0000 -> Channel 1 0111 -> Channel 81xxx -> ADCREFM
18-15	SEL_INM_SWM3_0	R/W	0h	SEL_INM pins for neg differential 0000 -> Channel 1 0111 -> Channel 81xxx -> ADCREFM
14-12	SEL_RFP__SWC_2_0	R/W	0h	SEL_RFP pins SW configuration 000 VDDA001- XPUL010 YPLL011 ADCREFP1xx - INTREF
11	WPNSW_SWC	R/W	0h	WPNSW pin SW configuration
10	YPNSW_SWC	R/W	0h	YPNSW pin SW configuration
9	XNPSW_SWC	R/W	0h	XNPSW pin SW configuration
8	YNNSW_SWC	R/W	0h	YNNSW pin SW configuration
7	YPPSW__SWC	R/W	0h	YPPSW pin SW configuration
6	XNNSW__SWC	R/W	0h	XNNSW pin SW configuration
5	XPPSW_SWC	R/W	0h	XPPSW pin SW configuration
4-0	Reserved		0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.51 TS_CHARGE_STEPCONFIG Register (offset = 44h) [reset = 0h]

TS Charge StepConfiguration@TSC_ADC_SS_TS_Charge_StepConfig Register

TS_CHARGE_STEPCONFIG is shown in Figure 11-55 and described in Table 11-56.

Figure 11-55. TS_CHARGE_STEPCONFIG Register

31	30	29	28	27	26	25	24
Reserved						Diff_CNTRL	SEL_RFM__SWC_1_0
-0h						R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
SEL_RFM__SWC_1_0	SEL_INP_SWC_3_0				SEL_INM_SWM3_0		
R/W-0h	R/W-0h				R/W-0h		

15	14	13	12	11	10	9	8
SEL_INM_SWM3_0	SEL_RFP__SWC_2_0			WPNSW_SWC	YPNSW_SWC	XNPSW_SWC	YNNSW_SWC
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
YPPSW__SWC	XNNSW__SWC	XPPSW_SWC	Reserved				
R/W-0h	R/W-0h	R/W-0h	-0h				

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 11-56. TS_CHARGE_STEPCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-26	Reserved		0h	
25	Diff_CNTRL	R/W	0h	Differential Control Pin 0 -> Single Ended 1 -> Differential Pair Enable
24-23	SEL_RFM__SWC_1_0	R/W	0h	SEL_RFM pins SW configuration 00 VSSA01- XNUR10 YNLR11 ADCREFM
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins SW configuration 0000 -> Channel 1 0111 -> Channel 81xxx -> ADCREFM
18-15	SEL_INM_SWM3_0	R/W	0h	SEL_INM pins for neg differential 0000 -> Channel 1 0111 -> Channel 81xxx -> ADCREFM
14-12	SEL_RFP__SWC_2_0	R/W	0h	SEL_RFP pins SW configuration 000 VDDA001- XPUL010 YPLL011 ADCREFP1xx - INTREF
11	WPNSW_SWC	R/W	0h	WPNSW pin SW configuration
10	YPNSW_SWC	R/W	0h	YPNSW pin SW configuration
9	XNPSW_SWC	R/W	0h	XNPSW pin SW configuration
8	YNNSW_SWC	R/W	0h	YNNSW pin SW configuration
7	YPPSW__SWC	R/W	0h	YPPSW pin SW configuration
6	XNNSW__SWC	R/W	0h	XNNSW pin SW configuration
5	XPPSW_SWC	R/W	0h	XPPSW pin SW configuration
4-0	Reserved		0h	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.52 TS_CHARGE_DELAY Register (offset = 48h) [reset = 1h]

TS Charge Delay Register@TSC_ADC_SS_TS_Charge_StepDelay Register

TS_CHARGE_DELAY is shown in Figure 11-56 and described in Table 11-57.

Figure 11-56. TS_CHARGE_DELAY Register

31	30	29	28	27	26	25	24
Reserved							
-0h							

23	22	21	20	19	18	17	16
Reserved						OpenDelay	
-0h						R/W-1h	

15	14	13	12	11	10	9	8
OpenDelay							
R/W-1h							

7	6	5	4	3	2	1	0
OpenDelay							
R/W-1h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-57. TS_CHARGE_DELAY Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved		0h	
17-0	OpenDelay	R/W	1h	Program the # of ADC clock cycles to wait between applying the step configuration registers and going back to the IDLE state. (Value must be > 0)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.53 STEPCONFIG1 Register (offset = 4Ch) [reset = 0h]

Step Configuration 1

STEPCONFIG1 is shown in Figure 11-57 and described in Table 11-58.

Figure 11-57. STEPCONFIG1 Register

31	30	29	28	27	26	25	24
Reserved				Range_check	FIFO_select	Diff_CNTRL	SEL_RFM_SWC_1_0
-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
SEL_RFM_SWC_1_0	SEL_INP_SWC_3_0				SEL_INM_SWM3_0		
R/W-0h	R/W-0h				R/W-0h		

15	14	13	12	11	10	9	8
SEL_INM_SWM3_0	SEL_RFP_SWC_2_0			WPNSW_SWC	YPNSW_SWC	XNPSW_SWC	YNNSW_SWC
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
YPPSW_SWC	XNNSW_SWC	XPPSW_SWC	Averaging			Mode	
R/W-0h	R/W-0h	R/W-0h	R/W-0h			R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-58. STEPCONFIG1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved		0h	
27	Range_check	R/W	0h	0 disable out of range check 1 - Compare adc data with range check register
26	FIFO_select	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 01 -> FIFO 1
25	Diff_CNTRL	R/W	0h	Differential Control Pin
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins SW configuration
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins SW configuration
18-15	SEL_INM_SWM3_0	R/W	0h	SEL_INP pins for neg differential
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins SW configuration
11	WPNSW_SWC	R/W	0h	WPNSW pin SW configuration
10	YPNSW_SWC	R/W	0h	YPNSW pin SW configuration
9	XNPSW_SWC	R/W	0h	XNPSW pin SW configuration
8	YNNSW_SWC	R/W	0h	YNNSW pin SW configuration
7	YPPSW_SWC	R/W	0h	YPPSW pin SW configuration
6	XNNSW_SWC	R/W	0h	XNNSW pin SW configuration
5	XPPSW_SWC	R/W	0h	XPPSW pin SW configuration
4-2	Averaging	R/W	0h	Number of samplings to average: 000 -> no average001 2 samples average010 4 samples average011 8 samples average 100 16 samples average
1-0	Mode	R/W	0h	00 SW enabled, one-shot01 SW enabled, continuous 10 HW synchronized, one-shot11 HW synchronized, continuous

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.54 IRQ_EOI Register (offset = 8h) [reset = 0h]

IRQ EOI

IRQ_EOI is shown in Figure 11-58 and described in Table 11-59.

Figure 11-58. IRQ_EOI Register

31	30	29	28	27	26	25	24
Reserved							
-0h							

23	22	21	20	19	18	17	16
Reserved							
-0h							

15	14	13	12	11	10	9	8
Reserved							
-0h							

7	6	5	4	3	2	1	0
Reserved							LINE_NUMBER
-0h							R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-59. IRQ_EOI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved		0h	
0	LINE_NUMBER	R/W	0h	Write 0: EOI for interrupt Read is always 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.55 STEPDELAY1 Register (offset = 50h) [reset = 0h]

Step Delay Register 1

STEPDELAY1 is shown in Figure 11-59 and described in Table 11-60.

Figure 11-59. STEPDELAY1 Register

31	30	29	28	27	26	25	24
SampleDelay							
R/W-0h							

23	22	21	20	19	18	17	16
Reserved						OpenDelay	
-0h						R/W-0h	

15	14	13	12	11	10	9	8
OpenDelay							
R/W-0h							

7	6	5	4	3	2	1	0
OpenDelay							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-60. STEPDELAY1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SampleDelay	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SOC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	Reserved		0h	
17-0	OpenDelay	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.56 STEPCONFIG2 Register (offset = 54h) [reset = 0h]

Step Configuration 2

STEPCONFIG2 is shown in Figure 11-60 and described in Table 11-61.

Figure 11-60. STEPCONFIG2 Register

31	30	29	28	27	26	25	24
Reserved				Range_check	FIFO_select	Diff_CNTRL	SEL_RFM_SWC_1_0
-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
SEL_RFM_SWC_1_0	SEL_INP_SWC_3_0				SEL_INM_SWM3_0		
R/W-0h	R/W-0h				R/W-0h		

15	14	13	12	11	10	9	8
SEL_INM_SWM3_0	SEL_RFP_SWC_2_0			WPNSW_SWC	YPNSW_SWC	XNPSW_SWC	YNNSW_SWC
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
YPPSW_SWC	XNNSW_SWC	XPPSW_SWC	Averaging			Mode	
R/W-0h	R/W-0h	R/W-0h	R/W-0h			R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-61. STEPCONFIG2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved		0h	
27	Range_check	R/W	0h	0 disable out of range check 1 - Compare adc data with range check register
26	FIFO_select	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 01 -> FIFO 1
25	Diff_CNTRL	R/W	0h	Differential Control Pin
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins SW configuration
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins SW configuration
18-15	SEL_INM_SWM3_0	R/W	0h	SEL_INP pins for neg differential
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins SW configuration
11	WPNSW_SWC	R/W	0h	WPNSW pin SW configuration
10	YPNSW_SWC	R/W	0h	YPNSW pin SW configuration
9	XNPSW_SWC	R/W	0h	XNPSW pin SW configuration
8	YNNSW_SWC	R/W	0h	YNNSW pin SW configuration
7	YPPSW_SWC	R/W	0h	YPPSW pin SW configuration
6	XNNSW_SWC	R/W	0h	XNNSW pin SW configuration
5	XPPSW_SWC	R/W	0h	XPPSW pin SW configuration
4-2	Averaging	R/W	0h	Number of samplings to average: 000 -> no average001 2 samples average010 4 samples average011 8 samples average 100 16 samples average
1-0	Mode	R/W	0h	00 SW enabled, one-shot01 SW enabled, continuous 10 HW synchronized, one-shot11 HW synchronized, continuous

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.57 STEPDELAY2 Register (offset = 58h) [reset = 0h]

Step Delay Register 2

STEPDELAY2 is shown in Figure 11-61 and described in Table 11-62.

Figure 11-61. STEPDELAY2 Register

31	30	29	28	27	26	25	24
SampleDelay							
R/W-0h							

23	22	21	20	19	18	17	16
Reserved						OpenDelay	
-0h						R/W-0h	

15	14	13	12	11	10	9	8
OpenDelay							
R/W-0h							

7	6	5	4	3	2	1	0
OpenDelay							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-62. STEPDELAY2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SampleDelay	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SOC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	Reserved		0h	
17-0	OpenDelay	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.58 STEPCONFIG3 Register (offset = 5Ch) [reset = 0h]

Step Configuration 3

STEPCONFIG3 is shown in Figure 11-62 and described in Table 11-63.

Figure 11-62. STEPCONFIG3 Register

31	30	29	28	27	26	25	24
Reserved				Range_check	FIFO_select	Diff_CNTRL	SEL_RFM_SWC_1_0
-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
SEL_RFM_SWC_1_0	SEL_INP_SWC_3_0				SEL_INM_SWM3_0		
R/W-0h	R/W-0h				R/W-0h		

15	14	13	12	11	10	9	8
SEL_INM_SWM3_0	SEL_RFP_SWC_2_0			WPNSW_SWC	YPNSW_SWC	XNPSW_SWC	YNNSW_SWC
R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
YPPSW_SWC	XNNSW_SWC	XPPSW_SWC	Averaging			Mode	
R/W-0h	R/W-0h	R/W-0h	R/W-0h			R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-63. STEPCONFIG3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-28	Reserved		0h	
27	Range_check	R/W	0h	0 disable out of range check 1 - Compare adc data with range check register
26	FIFO_select	R/W	0h	Sampled data will be stored in FIFO: 0 -> FIFO 01 -> FIFO 1
25	Diff_CNTRL	R/W	0h	Differential Control Pin
24-23	SEL_RFM_SWC_1_0	R/W	0h	SEL_RFM pins SW configuration
22-19	SEL_INP_SWC_3_0	R/W	0h	SEL_INP pins SW configuration
18-15	SEL_INM_SWM3_0	R/W	0h	SEL_INP pins for neg differential
14-12	SEL_RFP_SWC_2_0	R/W	0h	SEL_RFP pins SW configuration
11	WPNSW_SWC	R/W	0h	WPNSW pin SW configuration
10	YPNSW_SWC	R/W	0h	YPNSW pin SW configuration
9	XNPSW_SWC	R/W	0h	XNPSW pin SW configuration
8	YNNSW_SWC	R/W	0h	YNNSW pin SW configuration
7	YPPSW_SWC	R/W	0h	YPPSW pin SW configuration
6	XNNSW_SWC	R/W	0h	XNNSW pin SW configuration
5	XPPSW_SWC	R/W	0h	XPPSW pin SW configuration
4-2	Averaging	R/W	0h	Number of samplings to average: 000 -> no average001 2 samples average010 4 samples average011 8 samples average 100 16 samples average
1-0	Mode	R/W	0h	00 SW enabled, one-shot01 SW enabled, continuous 10 HW synchronized, one-shot11 HW synchronized, continuous

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11.5.59 STEPDELAY3 Register (offset = 60h) [reset = 0h]

Step Delay Register 3

STEPDELAY3 is shown in Figure 11-63 and described in Table 11-64.

Figure 11-63. STEPDELAY3 Register

31	30	29	28	27	26	25	24
SampleDelay							
R/W-0h							

23	22	21	20	19	18	17	16
Reserved						OpenDelay	
-0h						R/W-0h	

15	14	13	12	11	10	9	8
OpenDelay							
R/W-0h							

7	6	5	4	3	2	1	0
OpenDelay							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 11-64. STEPDELAY3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	SampleDelay	R/W	0h	This register will control the number of ADC clock cycles to sample (hold SOC high). Any value programmed here will be added to the minimum requirement of 1 clock cycle.
23-18	Reserved		0h	
17-0	OpenDelay	R/W	0h	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Chapter 12

SPRUH73–July 2011

LCD Controller

This chapter describes the LCD controller of the device.

Topic **Page**

12.1 Introduction **1514**
12.2 Integration **1516**
12.3 Functional Description **1518**
12.4 Use Cases **1539**
12.5 LCD Registers **1539**

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.1 Introduction

12.1.1 Purpose of the Peripheral

The LCD controller consists of two independent controllers, the Raster Controller and the LCD Interface Display Driver (LIDD) controller. Each controller operates independently from the other and only one of them is active at any given time.

- The Raster Controller handles the synchronous LCD interface. It provides timing and data for constant graphics refresh to a passive display. It supports a wide variety of monochrome and full-color display types and sizes by use of programmable timing controls, a built-in palette, and a gray-scale/serializer. Graphics data is processed and stored in frame buffers. A frame buffer is a contiguous memory block in the system. A built-in DMA engine supplies the graphics data to the Raster engine which, in turn, outputs to the external LCD device.
- The LIDD Controller supports the asynchronous LCD interface. It provides full-timing programmability of control signals (CS, WE, OE, ALE) and output data.

Figure 12-1 shows the LCD controller details. The raster and LIDD Controllers are responsible for generating the correct external timing. The DMA engine provides a constant flow of data from the frame buffer(s) to the external LCD panel via the Raster and LIDD Controllers. In addition, CPU access is provided to read and write registers.

The solid, thick lines in Figure 12-1 indicate the data path. The Raster Controller's data path is fairly complicated, for a thorough description of the Raster Controller data path, see Section 12.3.5.

Figure 12-1. LCD Controller

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.1.2 Features

General features of the LCD Controller include:

- Supports up to 24-bits data output; 8 bits-per-pixel (RGB).
- Supports up to WXGA resolution.
- Integrated DMA engine to pull data from the external frame buffer without burdening the processor via interrupts or a firmware timer.
- 512 word deep internal FIFO with programmable threshold values.
- Character Based Panels
  - Supports 2 Character Panels (CS0 and CS1) with independent and programmable bus timing parameters when in asynchronous Hitachi, Motorola and Intel modes
  - Supports 1 Character Panel (CS0) with programmable bus timing parameters when in synchronous Motorola and Intel modes
  - Can be used as a generic 16 bit address/data interleaved MPU bus master with no external stall
- Passive Matrix LCD Panels
  - Panel types including STN, DSTN, and C-DSTN
  - AC Bias Control
- Active Matrix LCD Panels
  - Panel types including TN TFT
- OLED Panels
  - Passive Matrix (PM OLED) with frame buffer and controller IC inside the Panel
  - Active Matrix (AM OLED)

12.1.3 Unsupported Features

The LCD Controller does not support resolutions above WXGA due to IO timing limitations.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.2 Integration

The device includes an LCD Controller that reads display data from external memory and drives several different types of LCD displays. The LCD Controller integration is shown in Figure 12-2.

Figure 12-2. LCD Controller Integration

12.2.1 LCD Controller Connectivity Attributes

The general connectivity attributes for the LCDC subsystems are shown in Table 12-1.

Table 12-1. LCD Controller Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_LCD_L3_GCLK (OCP Master Clock) PD_PER_LCD_L3_GCLK (OCP Slave Clock) PD_PER_LCD_GCLK (Functional Clock)
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Standby Smart Idle
Interrupt Requests	1 Interrupt to MPU Subsystem
DMA Requests	None
Physical Address	L4 Peripheral Slave Port

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.2.2 LCD Controller Clock and Reset Management

The LCDC module uses the following functional and OCP interface clocks. The L4 Slave interface runs at half the frequency of the L3 Master interface but uses the same clock. The clock is divided within the LCDC through the l4_clkdiv input using the pd_per_lcd_l4s_gclk_ien signal from the PRCM. The functional clock comes from the Display PLL. When the Display PLL is in bypass mode, it's output is sourced by either CORE_CLKOUTM6 or PER_CLKOUTM2.

Table 12-2. LCD Controller Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
l3_clk Master Interface Clock	200 MHz	CORE_CLKOUTM4	pd_per_lcd_l3_gclk From PRCM
l4_clk Slave Interface Clock	100 MHz	CORE_CLKOUTM4 (divided within LCDC)	pd_per_lcd_l3_gclk From PRCM
lcd_clk Functional Clock	200 MHz	Display PLL CLKOUT	pd_per_lcd_gclk From PRCM

12.2.3 LCD Controller Pin List

The LCD Controller external interface signals are shown in Table 12-3.

Table 12-3. LCD Controller Pin List

Pin	Type	Description
lcd_cp	O	Pixel Clock in Raster model Read Strobe or Read/Write Strobe in LIDD mode
lcd_pixel_i[15:0]	I	LCD Data Bus input (for LIDD mode only)
lcd_pixel_o[23:0]	O	LCD Data Bus output
lcd_lp	O	Line Clock or HSYNC in Raster mode; Write Strobe or Direction bit in LIDD mode
lcd_fp	O	Frame Clock or VSYNC in Raster mode; Address Latch Enable in LIDD mode
lcd_ac	O	AC bias or Latch Enable in Raster mode; Primary Chip Select/Primary Enable in LIDD MPU/Hitachi mode
lcd_mclk	O	N/A in Raster mode; Memory Clock/Secondary chip Select/Secondary Enable in LIDD Synchronous/Async MPU/Hitachi mode

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.3 Functional Description

12.3.1 Clocking

This section details the various clocks and signals. Figure 12-3 shows input and output LCD controller clocks.

Figure 12-3. Input and Output Clocks

12.3.1.1 Pixel Clock

The pixel clock (LCD_PCLK) frequency is derived from LCD_CLK, the reference clock to this LCD module (see Figure 12-3). The pixel clock is used by the LCD display to clock the pixel data into the line shift register.

$$\text{LCD_PCLK} = \frac{\text{LCD_CLK}}{\text{CLKDIV}}$$

where CLKDIV is a field in the LCD_CTRL register and should not be 0 or 1.

- **Passive (STN) mode.** LCD_PCLK only transitions when valid data is available for output. It does not transition when the horizontal clock is asserted or during wait state insertion.
- **Active (TFT) mode.** LCD_PCLK continuously toggles as long as the Raster Controller is enabled.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.3.1.2 Horizontal Clock (LCD_HSYNC)

LCD_HSYNC toggles after all pixels in a horizontal line have been transmitted to the LCD and a programmable number of pixel clock wait states has elapsed both at the beginning and end of each line.

The RASTER_TIMING_0 register fully defines the behavior of this signal.

LCD_HSYNC can be programmed to be synchronized with the rising or falling edge of LCD_PCLK. The configuration field is bits 24 and 25 in the RASTER_TIMING_2 register.

- **Active (TFT) mode.** The horizontal clock or the line clock is also used by TFT displays as the horizontal synchronization signal (LCD_HSYNC).

  The timings of the horizontal clock(line clock) pins are programmable to support:
- Delay insertion both at the beginning and end of each line
- Line clock polarity.
- Line clock pulse width, driven on rising or falling edge of pixel clock.

12.3.1.3 Vertical Clock (LCD_VSYNC)

LCD_VSYNC toggles after all lines in a frame have been transmitted to the LCD and a programmable number of line clock cycles has elapsed both at the beginning and end of each frame.

The RASTER_TIMING_1 register fully defines the behavior of this signal.

LCD_VSYNC can be programmed to be synchronized with the rising or falling edge of LCD_PCLK. The configuration field is bits 24 and 25 in the RASTER_TIMING_2 register.

- **Passive (STN) mode.** The vertical clock; that is, the frame clock, toggles during the first line of the screen.
- **Active (TFT) mode.** The vertical clock, that is, the frame clock, is also used by TFT displays as the vertical synchronization signal (LCD_VSYNC).

  The timings of the vertical clock pins are programmable to support:
- Delay insertion both at the beginning and end of each frame
- Frame clock polarity

12.3.1.4 $\overline{\text{LCD_AC_ENB_CS}}$

- **Passive (STN) mode.** To prevent a dc charge within the screen pixels, the power and ground supplies of the display are periodically switched. The Raster Controller signals the LCD to switch the polarity by toggling this pin ($\overline{\text{LCD_AC_ENB_CS}}$).
- **Active (TFT) mode.** This signal acts as an output enable (OE) signal. It is used to signal the external LCD that the data is valid on the data bus (LCD_D[15:0]). (see Figure 12-25)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.3.2 LCD External I/O Signals

Table 12-4 shows the details of the LCD controller external signals.

Table 12-4. LCD External I/O Signals

Signal	Type	Description
LCD_VSYNC	OUT	Raster controller: Frame clock the LCD uses to signal the start of a new frame of pixels. Also used by TFT displays as the vertical synchronization signal. LIDD character: Register select (RS) or address latch enable (ALE) LIDD graphics: Address bit 0 (A0) or command/data select (C/D)
LCD_HSYNC	OUT	Raster controller: Line clock the LCD uses to signal the end of a line of pixels that transfers line data from the shift register to the screen and to increment the line pointer(s). Also used by TFT displays as the horizontal synchronization signal. LIDD character: not used. LIDD graphics: • 6800 mode = read or write enable • 8080 mode = not write strobe
LCD_PCLK	OUT	Raster controller: Pixel clock the LCD uses to clock the pixel data into the line shift register. In passive mode, the pixel clock transitions only when valid data is available on the data lines. In active mode, the pixel clock transitions continuously, and the ac-bias pin is used as an output enable to signal when data is available on the LCD pin. LIDD character: not used. LIDD graphics: • 6800 mode = enable strobe • 8080 mode = not read strobe
$\overline{\text{LCD_AC_ENB_CS}}$	OUT	Raster controller: ac-bias used to signal the LCD to switch the polarity of the power supplies to the row and column axis of the screen to counteract DC offset. Used in TFT mode as the output enable to signal when data is latched from the data pins using the pixel clock. LIDD character: Primary enable strobe LIDD graphics: Chip select 0 (CS0)
LCD_MCLK	OUT	Raster controller: not used. LIDD character: Secondary enable strobe LIDD graphics: Chip select 1 (CS1)
LCD_D[15:0]	Raster: OUT	LCD data bus, providing a 4-, 8-, or 16-bit data path.
	LIDD: OUT/IN	Raster controller: For monochrome displays, each signal represents a pixel; for passive color displays, groupings of three signals represent one pixel (red, green, and blue). LCD_D[3:0] is used for monochrome displays of 2, 4, and 8 BPP; LCD_D[7:0] is used for color STN displays and LCD_D[15:0] is used for active (TFT) mode. LIDD character: Read and write the command and data registers. LIDD graphics: Read and write the command and data registers.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.3.3 DMA Engine

The DMA engine provides the capability to output graphics data to constantly refresh LCDs, without burdening the CPU, via interrupts or a firmware timer. It operates on one or two frame buffers, which are set up during initialization. Using two frame buffers (ping-pong buffers) enables the simultaneous operation of outputting the current video frame to the external display and updating the next video frame. The ping-pong buffering approach is preferred in most applications.

When the Raster Controller is used, the DMA engine reads data from a frame buffer and writes it to the input FIFO (as shown in Figure 12-1). The Raster Controller requests data from the FIFO for frame refresh; as a result, the DMA's job is to ensure that the FIFO is always kept full.

When the LIDD Controller is used, the DMA engine accesses the LIDD Controller's address and/or data registers.

To program DMA engine, configure the following registers, as shown in Table 12-5.

Table 12-5. Register Configuration for DMA Engine Programming

Register	Configuration
LCDDMA_CTRL	Configure DMA data format
LCDDMA_FB0_BASE LCDDMA_FB0_CEILING	Configure frame buffer 0
LCDDMA_FB1_BASE LCDDMA_FB1_CEILING	Configure frame buffer 1. (If only one frame buffer is used, these two registers will not be used.)

In addition, the LIDD_CTRL register (for LIDD Controller) or the RASTER_CTRL register (for Raster Controller) should also be configured appropriately, along with all the timing registers.

To enable DMA transfers, the LIDD_DMA_EN bit (in the LIDD_CTRL register) or the LCDEN bit (in the RASTER_CTRL register) should be written with 1.

12.3.3.1 Interrupts

Interrupts in this LCD module are related to DMA engine operation. Three registers are closely related to this subject:

- The LIDD_CTRL and RASTER_CTRL registers enable or disable each individual interrupt sources.
- The LCD_STAT register collects all the interrupt status information.

12.3.3.1.1 LIDD Mode

When operating in LIDD mode, the DMA engine generates one interrupt signal every time the specified frame buffer has been transferred completely.

- The DONE_INT_EN bit in the LIDD_CTRL register specifies if the interrupt signal is delivered to the system interrupt controller, which in turn may or may not generate an interrupt to CPU.
- The EOF1, EOF0, and DONE bits in the LCD_STAT register reflect the interrupt signal, regardless of being delivered to the system interrupt controller or not.

12.3.3.1.2 Raster Mode

When operating in Raster mode, the DMA engine can generate the interrupts in the following scenarios:

1. **Output FIFO under-run**. This occurs when the DMA engine cannot keep up with the data rate consumed by the LCD (which is determined by the LCD_PCLK.) This is likely due to a system memory throughput issue or an incorrect LCD_PCLK setting. The FUF bit in LCD_STAT is set when this error occurs. This bit is cleared by disabling the Raster Controller (i.e., clearing the LCDEN bit in RASTER_CTRL).

Copyright © 2011, Texas Instruments Incorporated

2. **Frame synchronization lost**. This error happens when the DMA engine attempts to read what it believes to be the first word of the video buffer but it cannot be recognized as such. This could be caused by an invalid frame buffer address or an invalid BPP value (for more details, see Section 12.3.5.2). The SYNC bit in the LCD_STAT register is set when such an error is detected. This field is cleared by disabling the Raster Controller (clearing the LCDEN bit in the RASTER_CTRL register).
3. **Palette loaded**. This interrupt can be generated when the palette is loaded into the memory by the DMA engine. At the same time, the PL bit in the LCD_STAT register is set. In data-only (PLM = 2h) and palette-plus-data (PLM = 00) modes, writing 0 to this bit clears the interrupt. In the palette-only (PLM = 1) mode, this bit is cleared by disabling the Raster Controller (clearing the LCDEN bit in the RASTER_CTRL register).
4. **AC bias transition**. If the ACB_I bit in the RASTER_TIMING_2 register is programmed with a non-zero value, an internal counter will be loaded with this value and starts to decrement each time $\overline{\text{LCD_AC_ENB_CS}}$ (AC-bias signal) switches its state. When the counter reaches zero, the ABC bit in the LCD_STAT register is set, which will deliver an interrupt signal to the system interrupt controller (if the interrupt is enabled.) The counter reloads the value in field ACB_I, but does not start to decrement until the ABC bit is cleared by writing 0 to this bit.
5. **Frame transfer completed**. When one frame of data is transferred completely, the DONE bit in the LCD_STAT register is set. This bit is cleared by disabling the Raster Controller (i.e., clearing the LCDEN bit in the RASTER_CTRL register). Note that the EOF0 and EOF1 bits in the LCD_STAT register will be set accordingly.

Note that the interrupt enable bits are in the RASTER_CTRL register. The corresponding enable bit must be set in order to generate an interrupt to the CPU. However, the LCD_STAT register reflects the interrupt signal regardless of the interrupt enable bits settings.

12.3.3.1.3 Interrupt Handling

Refer the device-specific data manual for information about LCD interrupt number to CPU .The interrupt service routine needs to determine the interrupt source by examining the LCD_STAT register and clearing the interrupt properly.

12.3.4 LIDD Controller

The LIDD Controller is designed to support LCD panels with a memory-mapped interface. The types of displays range from low-end character monochrome LCD panels to high-end TFT smart LCD panels.

LIDD mode (and the use of this logic) is enabled by clearing the MODESEL bit in the LCD control register (LCD_CTRL).

LIDD Controller operation is summarized as follows:

- During initialization, the LCD LIDD CS0/CS1 configuration registers (LIDD_CS0_CONF and LIDD_CS1_CONF) are configured to match the requirements of the LCD panel being used.
- During normal operation, the CPU writes display data to the LCD data registers (LIDD_CS0_DATA and LIDD_CS1_DATA). The LIDD interface converts the CPU write into the proper signal transition sequence for the display, as programmed earlier. Note that the first CPU write should send the beginning address of the update to the LCD panel and the subsequent writes update data at display locations starting from the first address and continuing sequentially. Note that DMA may be used instead of CPU.
- The LIDD Controller is also capable of reading back status or data from the LCD panel, if the latter has this capability. This is set up and activated in a similar manner to the write function described above.

NOTE: If an LCD panel is not used, this interface can be used to control any MCU-like peripheral.

See your device-specific data manual to check the LIDD features supported by the LCD controller.

Table 12-6 describes how the signals are used to interface external LCD modules, which are configured by the LIDD_CTRL register.

Copyright © 2011, Texas Instruments Incorporated

Table 12-6. LIDD I/O Name Map

Display Type	Interface Type	Data Bits	LIDD_CTRL [2:0]	I/O Name	Display I/O Name	Comment
Character Display	HD44780 Type	4	100	LCD_D[7:4]	DATA[7:4]	Data Bus (length defined by Instruction)
				LCD_PCLK	E (or E0)	Enable Strobe (first display)
				LCD_HSYNC	R/$\overline{W}$	Read/$\overline{Write}$
				LCD_VSYNC	RS	Register Select (Data/not Instruction)
				LCD_MCLK	E1	Enable Strobe (second display optional)
Character Display	HD44780 Type	8	100	LCD_D[7:0]	DATA[7:0]	Data Bus (length defined by Instruction)
				LCD_PCLK	E (or E0)	Enable Strobe (first display)
				LCD_HSYNC	R/$\overline{W}$	Read/$\overline{Write}$
				LCD_VSYNC	RS	Register Select (Data/not Instruction)
				LCD_MCLK	E1	Enable Strobe (second display optional)
Micro Interface Graphic Display	6800 Family	Up to 16	001	LCD_D[15:0]	DATA[7:0]	Data Bus (16 bits always available)
				LCD_PCLK	E	Enable Clock
				LCD_HSYNC	R/$\overline{W}$	Read/$\overline{Write}$
				LCD_VSYNC	A0	Address/Data Select
				$\overline{LCD_AC_ENB_CS}$	CS (or CS0)	Chip Select (first display)
				LCD_MCLK	CS1	Chip Select (second display optional)
			000	LCD_MCLK	None	Synchronous Clock (optional)
Micro Interface Graphic Display	8080 Family	Up to 16	011	LCD_D[15:0]	DATA[7:0]	Data Bus (16 bits always available)
				LCD_PCLK	RD	Read Strobe
				LCD_HSYNC	WR	Write Strobe
				LCD_VSYNC	A0	Address/Data Select
				$\overline{LCD_AC_ENB_CS}$	CS (or CS0)	Chip Select (first display)
				LCD_MCLK	CS1	Chip Select (second display optional)
			010	LCD_MCLK	None	Synchronous Clock (optional)

The timing parameters are defined by the LIDD_CS0_CONF and LIDD_CS1_CONF registers, which are described in Section 12.5.7.

The timing configuration is based on an internal reference clock, MCLK. The MCLK is generated out of LCD_CLK, which is determined by the CLKDIV bit in the LCD_CTRL register.

$$\text{MCLK} = \text{LCD_CLK when CLKDIV} = 0.$$

$$\text{MCLK} = \frac{\text{LCD_CLK}}{\text{CLKDIV}} \text{ when CLKDIV} \neq 0.$$

See your device-specific data manual for the timing configurations supported by the LCD controller.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.3.4.1 LIDD Controller Timing

Figure 12-4. LIDD Mode HD44780 Write Timing Diagram

Figure 12-5. LIDD Mode HD44780 Read Timing Diagram

Figure 12-6. LIDD Mode 6800 Write Timing Diagram

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 12-7. LIDD Mode 6800 Read Timing Diagram

Figure 12-8. LIDD Mode 6800 Status Timing Diagram

Figure 12-9. LIDD Mode 8080 Write Timing Diagram

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 12-10. LIDD Mode 8080 Read Timing Diagram

Figure 12-11. LIDD Mode 8080 Status Timing Diagram

NOTE: The CS_DELAY in above figures is same as bit field TA in

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.3.5 Raster Controller

Raster mode (and the use of this logic) is enabled by setting the MODESEL bit in the LCD control register (LCD_CTRL). Table 12-7 shows the active external signals when this mode is active.

Table 12-7. Operation Modes Supported by Raster Controller

Interface	Data Bus Width	Register Bits RASTER_CTRL[9, 7, 1]	Signal Name	Description
Passive (STN) Mono 4-bit	4	001	LCD_D[3:0]	Data bus
			LCD_PCLK	Pixel clock
			LCD_HSYNC	Horizontal clock(Line Clock)
			LCD_VSYNC	Vertical clock (Frame Clock)
			$\overline{\text{LCD_AC_ENB_CS}}$	AC Bias
			LCD_MCLK	Not used
Passive (STN) Mono 8-bit	8	101	LCD_D[7:0]	Data bus
			LCD_PCLK	Pixel clock
			LCD_HSYNC	Horizontal clock(Line Clock)
			LCD_VSYNC	Vertical clock (Frame Clock)
			$\overline{\text{LCD_AC_ENB_CS}}$	AC Bias
			LCD_MCLK	Not used
Passive (STN) Color	8	100	LCD_D[7:0]	Data bus
			LCD_PCLK	Pixel clock
			LCD_HSYNC	Horizontal clock(Line Clock)
			LCD_VSYNC	Vertical clock (Frame Clock)
			$\overline{\text{LCD_AC_ENB_CS}}$	AC Bias
			LCD_MCLK	Not used
Active (TFT) Color	16	x10	LCD_D[15:0]	Data bus
			LCD_PCLK	Pixel clock
			LCD_HSYNC	Horizontal clock(Line Clock)
			LCD_VSYNC	Vertical clock (Frame Clock)
			$\overline{\text{LCD_AC_ENB_CS}}$	Output enable
			LCD_MCLK	Not used

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.3.5.1 Logical Data Path

The block diagram of the Raster Controller is shown in Figure 12-1. Figure 12-12 illustrates its logical data path for various operation modes (passive (STN) versus active (TFT), various BPP size).

Figure 12-12 shows that:

- The gray-scaler/serializer and output FIFO blocks are bypassed in active (TFT) modes.
- The palette is bypassed in both 12- and 16-BPP modes.

Figure 12-12. Logical Data Path for Raster Controller

In summary:

- The display image is stored in frame buffers.
- The built-in DMA engine constantly transfers the data stored in the frame buffers to the Input FIFO.
- The Raster Controller relays data to the external pins according to the specified format.

The remainder of this section describes the functioning blocks in Figure 12-12, including frame buffers, palette, and gray-scaler/serializer. Their operation and programming techniques are covered in detail. The output format is also described in Section 12.3.5.5.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.3.5.2 Frame Buffer

A frame buffer is a contiguous memory block, storing enough data to fill a full LCD screen. For this device, external memory needs to be used for the frame buffer. For specific details on which external memory interface (EMIF) controller can be accessed by the LCD controller, see your device-specific data manual. The data in the frame buffer consists of pixel values as well as a look-up palette. Figure 12-13 shows the frame buffer structure.

Figure 12-13. Frame Buffer Structure

1, 2, 4, 12, 16 BPP Modes

8 BPP Mode

NOTE:

- 8-BPP mode uses the first 512 bytes in the frame buffer as the palette while the other modes use 32 bytes.
- 12- and 16-BPP modes do not need a palette; i.e., the pixel data is the desired RGB value. However, the first 32 bytes are still considered a palette. The first entry should be 4000h (bit 14 is 1) while the remaining entries must be filled with 0. (For details, see Table 12-8.)
- Each entry in a palette occupies 2 bytes. As a result, 8-BPP mode palette has 256 color entries while the other palettes have up to 16 color entries.
- 4-BPP mode uses up the all the 16 entries in a palette.
- 1-BPP mode uses the first 2 entries in a palette while 2-BPP mode uses the first 4 entries. The remaining entries are not used and must be filled with 0.
- In 12- and 16-BPP modes, pixel data is RGB data. For all the other modes, pixel data is actually an index of the palette entry.

Table 12-8. Bits-Per-Pixel Encoding for Palette Entry 0 Buffer

Bit	Name	Value	Description [(1)] [(2)]
14-12	BPP		Bits-per-pixel.
		000	1 BPP
		001	2 BPP
		010	4 BPP
		011	8 BPP
		1xx	12 BPP in passive mode (TFT_STN = 0 and STN_565 = 0 in RASTER_CTRL)
			16 BPP in passive mode (TFT_STN = 0 and STN_565 = 1 in RASTER_CTRL)
			16 BPP in active mode (TFT_STN = 1 in RASTER_CTRL)

(1) Eight 1-bit pixels, four 2-bit pixels, and two 4-bit pixels are packed into each byte, and 12-bit pixels are right justified on (16-bit) word boundaries (in the same format as palette entry).

(2) For STN565, see the 16 BPP STN mode bit ().

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

The equations shown in Table 12-9 are used to calculate the total frame buffer size (in bytes) based on varying pixel size encoding and screen sizes.

Figure 12-14 and Figure 12-15 show more detail of the palette entry organization.

Table 12-9. Frame Buffer Size According to BPP

BPP	Frame Buffer Size
1	32 + (Lines × Columns)/8
2	32 + (Lines × Columns)/4
4	32 + (Lines × Columns)/2
8	512 + (Lines × Columns)
12/16	32 + 2 × (Lines × Columns)

Figure 12-14. 16-Entry Palette/Buffer Format (1, 2, 4, 12, 16 BPP)

A. Bits-per-pixels (BPP) is only contained within the first palette entry (palette entry 0).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 12-15. 256-Entry Palette/Buffer Format (8 BPP)

Individual Palette Entry

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Color	Unused	BPP[A]			Red (R)				Green (G)				Blue (B)			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Mono	Unused	BPP[A]			Unused								Mono (M)			

256-Entry Palette Buffer

Bit	15 … 0
Base + 0	Palette Entry 0
Base + 2	Palette Entry 1
. . .	. . .
Base + FCh	Palette Entry 254
Base + FEh	Palette Entry 255

A. Bits-per-pixels (BPP) is only contained within the first palette entry (palette entry 0).

Bits 12, 13, and 14 of the first palette entry select the number of bits-per-pixel to be used in the following frame and thus the number of palette RAM entries. The palette entry is used by the Raster Controller to correctly unpack pixel data.

The following figures show the memory organization within the frame buffer for each pixel encoding size.

Figure 12-16. 16-BPP Data Memory Organization (TFT Mode Only)—Little Endian

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
16 bits/pixel	R					G						B				

Bit	15 … 0
Base	Pixel 0
Base + 2	Pixel 1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 12-17. 12-BPP Data Memory Organization—Little Endian

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
12 bits/pixel	Unused				R				G				B			

Bit	15	0
Base	Pixel 0	
Base + 2	Pixel 1	

Unused [15-12] bits are filled with zeroes in TFT mode.

Figure 12-18. 8-BPP Data Memory Organization

Bit	7	6	5	4	3	2	1	0
8 bits/pixel	Data[7:0]							

Bit	7	0
Base	Pixel 0	
Base + 1	Pixel 1	
Base + 2	Pixel 2	

Figure 12-19. 4-BPP Data Memory Organization

Bit	3	2	1	0
4 bits/pixel	Data[3:0]			

Bit	7 4	3 0
Base	Pixel 0	Pixel 1
Base + 1	Pixel 2	Pixel 3
Base + 2	Pixel 4	Pixel 5

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 12-20. 2-BPP Data Memory Organization

Bit	7 6	5 4	3 2	1 0
Base	Pixel 0	Pixel 1	Pixel 2	Pixel 3
Base + 1	Pixel 4	Pixel 5	Pixel 6	Pixel 7
Base + 2	Pixel 8	Pixel 9	Pixel 10	Pixel 11

Figure 12-21. 1-BPP Data Memory Organization

Bit	7	6	5	4	3	2	1	0
Base	P0	P1	P2	P3	P4	P5	P6	P7
Base + 1	P8	P9	P10	P11	P12	P13	P14	P15

12.3.5.3 Palette

As explained in the previous section, the pixel data is an index of palette entry (when palette is used). The number of colors supported is given by $2^{\text{number of BPP}}$. However, due to a limitation of the gray-scaler/serializer block, fewer grayscales or colors may be supported.

The PLM field (in RASTER_CTRL) affects the palette loading:

- If PLM is 00b (palette-plus-data mode) or 01b (palette-only mode), the palette is loaded by the DMA engine at the very beginning, which is followed by the loading of pixel data.
- If PLM is 10b (data-only mode), the palette is not loaded. Instead, the DMA engine loads the pixel data immediately.

12.3.5.4 Gray-Scaler/Serializer

12.3.5.4.1 Passive (STN) Mode

Once a palette entry is selected from the look-up palette by the pixel data, its content is sent to the gray-scaler/serializer. If it is monochrome data, it is encoded as 4 bits. If it is color data, it is encoded as 4 bits (Red), 4 bits (Green), and 4 bits (Blue).

These 4-bit values are used to select one of the 16 intensity levels, as shown in Table 12-10. A patented algorithm is used during this processing to provide an optimized intensity value that matches the eye's visual perception of color/gray gradations.

12.3.5.4.2 Active (TFT) Mode

The gray-scaler/serializer is bypassed.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 12-10. Color/Grayscale Intensities and Modulation Rates

Dither Value (4-Bit Value from Palette)	Intensity (0% is White)	Modulation Rate (Ratio of ON to ON+OFF Pixels)
0000	0.0%	0
0001	11.1%	1/9
0010	20.0%	1/5
0011	26.7%	4/15
0100	33.3%	3/9
0101	40.0%	2/5
0110	44.4%	4/9
0111	50.0%	1/2
1000	55.6%	5/9
1001	60.0%	3/5
1010	66.6%	6/9
1011	73.3%	11/15
1100	80.0%	4/5
1101	88.9%	8/9
1110	100.0%	1
1111	100.0%	1

12.3.5.4.3 Summary of Color Depth

Table 12-11. Number of Colors/Shades of Gray Available on Screen

Number of BPP	Passive Mode (TFT_STN = 0)		Active Mode (TFT_STN = 1)
	Monochrome (MONO_COLOR = 1)	Color (MONO_COLOR = 0)	Color Only (MONO_COLOR = 0)
1	2 palette entries to select within 15 grayscales	2 palette entries to select within 3375 possible colors	2 palette entries to select within 4096 possible colors
2	4 palette entries to select within 15 grayscales	4 palette entries to select within 3375 possible colors	4 palette entries to select within 4096 possible colors
4	16 palette entries to select within 15 grayscales	16 palette entries to select within 3375 possible colors	16 palette entries to select within 4096 possible colors
8	Not relevant since it would consist in 256 palette entries to select within 15 grayscales, but exists anyway	256 palette entries to select 3375 possible colors	256 palette entries to select within 4096 possible colors
12	x	3375 possible colors	4096 possible colors
16	x	3375 possible colors (STN_565 = 1)	Up to 65536 possible colors

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.3.5.5 Output Format

12.3.5.5.1 Passive (STN) Mode

As shown in Figure 12-12, the pixel data stored in frame buffers go through palette (if applicable) and gray-scaler/serializer before reaching the Output FIFO. As a result, it is likely that the data fed to the Output FIFO is numerically different from the data in the frame buffers. (However, they represent the same color or grayscale.)

The output FIFO formats the received data according to display modes (see Table 12-7). Figure 12-22 shows the actual data output on the external pins.

12.3.5.5.2 Active (TFT) Mode

As shown in Figure 12-12, the gray-scaler/serializer and output FIFO are bypassed in active (TFT) mode. Namely, at each pixel clock, one pixel data (16 bits) is output to the external LCD.

Figure 12-22. Monochrome and Color Output

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.3.5.6 Subpanel Feature

In some applications, it is desired to display only the first or last few lines of the LCD panel (see Figure 12-23). This is mainly used for power saving.

This is supported by the Raster Controller via its subpanel feature. The RASTER_SUBPANEL register fully defines its behavior, that is, the following parameters are defined:

- Whether the first or last few lines will be refreshed.
- A line number, which is the last (or first) line to be refreshed.
- The pixel data to be loaded to the refresh area.

Note that there is only one pixel value for all the pixels in the refresh area. As a result, frame buffers and DMA engine are not used in this case, which leads to power saving.

Figure 12-23. Raster Mode Display Format

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.3.5.7 Raster Controller Timing

Figure 12-24. Raster Mode Passive (STN) Timing Diagram

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 12-25. Raster Mode Active (TFT) Timing Diagram

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.4 Use Cases

12.5 LCD Registers

Table 12-12 lists the memory-mapped registers for the LCD. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 12-12 should be considered as reserved locations and the register contents should not be modified.

Table 12-12. LCD REGISTERS

Offset	Acronym	Register Description	Section
0h	pid		Section 12.5.1
64h	IRQENABLE_CLEAR		Section 12.5.2
68h	IRQEOI_VECTOR		Section 12.5.3
6Ch	CLKC_ENABLE		Section 12.5.4
70h	CLKC_RESET		Section 12.5.5
Ch	LIDD_CTRL		Section 12.5.6
10h	LIDD_CS0_CONF		Section 12.5.7
14h	LIDD_CS0_ADDR		Section 12.5.8
E8h	reg29		Section 12.5.9
ECh	reg30		Section 12.5.10
18h	LIDD_CS0_DATA		Section 12.5.11
F0h	reg31		Section 12.5.12
1Ch	LIDD_CS1_CONF		Section 12.5.13
20h	LIDD_CS1_ADDR		Section 12.5.14
24h	LIDD_CS1_DATA		Section 12.5.15
4h	CTRL		Section 12.5.16
28h	RASTER_CTRL		Section 12.5.17
2Ch	RASTER_TIMING_0		Section 12.5.18
30h	RASTER_TIMING_1		Section 12.5.19
34h	RASTER_TIMING_2		Section 12.5.20
38h	RASTER_SUBPANEL		Section 12.5.21
3Ch	RASTER_SUBPANEL2		Section 12.5.22
40h	LCDDMA_CTRL		Section 12.5.23
44h	LCDDMA_FB0_BASE		Section 12.5.24
48h	LCDDMA_FB0_CEILING		Section 12.5.25
4Ch	LCDDMA_FB1_BASE		Section 12.5.26
50h	LCDDMA_FB1_CEILING		Section 12.5.27
54h	SYSCONFIG		Section 16.2.2.2
58h	IRQSTATUS_RAW		Section 12.5.29
5Ch	IRQSTATUS		Section 12.5.30
60h	IRQENABLE_SET		Section 12.5.31

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.1 pid Register (offset = 0h) [reset = 4F200000h]

pid is shown in Figure 12-26 and described in Table 12-13.

Figure 12-26. pid Register

31	30	29	28	27	26	25	24
scheme		Reserved		func			
R-1h				R-F20h			

23	22	21	20	19	18	17	16
func							
R-F20h							

15	14	13	12	11	10	9	8
rtl					major		
R-0h					R-0h		

7	6	5	4	3	2	1	0
custom		minor					
R-0h		R-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-13. pid Register Field Descriptions

Bit	Field	Type	Reset	Description
31-30	scheme	R	1h	The scheme of the register used. This field indicates the 3.5 Method.
29-28	Reserved			
27-16	func	R	F20h	The function of the module being used.
15-11	rtl	R	0h	The Release number for this IP.
10-8	major	R	0h	Major Release Number
7-6	custom	R	0h	Custom IP
5-0	minor	R	0h	Minor Release Number

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.2 IRQENABLE_CLEAR Register (offset = 64h) [reset = 0h]

IRQENABLE_CLEAR is shown in Figure 12-27 and described in Table 12-14.

Figure 12-27. IRQENABLE_CLEAR Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved						eof1_en_clr	eof0_en_clr
						R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
Reserved	pl_en_clr	fuf_en_clr	Reserved	acb_en_clr	sync_en_clr	Reserved	done_en_clr
	R/W-0h	R/W-0h		R/W-0h	R/W-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-14. IRQENABLE_CLEAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	Reserved			
9	eof1_en_clr	R/W	0h	DMA End-of-Frame 1 Interrupt Enable Clear Read indicates enabled status 0 = disabled 1 = enabled Writing 1 will clear interrupt enable Writing 0 has no effect
8	eof0_en_clr	R/W	0h	DMA End-of-Frame 0 Interrupt Enable ClearRead indicates enabled status 0 = disabled 1 = enabledWriting 1 will clear interrupt enableWriting 0 has no effect
7	Reserved			
6	pl_en_clr	R/W	0h	DMA Palette Loaded Interrupt Enable ClearRead indicates enabled status 0 = disabled 1 = enabledWriting 1 will clear interrupt enableWriting 0 has no effect
5	fuf_en_clr	R/W	0h	DMA FIFO Underflow Interrupt Enable Clear LCD dithering logic not supplying data to FIFO at a sufficient rate, FIFO has completely emptied and data pin driver logic has attempted to take added data from FIFO. Read indicates enabled status 0 = disabled 1 = enabled Writing 1 will clear interrupt enable Writing 0 has no effect
4	Reserved			
3	acb_en_clr	R/W	0h	For Passive Matrix Panels OnlyAC Bias Count Interrupt Enable ClearAC bias transition counter has decremented to zero, indicating that the lcd_ac_o line has transitioned the number of times which is specified by the acbi control bit-field. The counter is reloaded with the value in acbi but it is disabled until the user clears ABC.Read indicates enabled status 0 = disabled 1 = enabledWriting 1 will clear interrupt enableWriting 0 has no effect
2	sync_en_clr	R/W	0h	Frame Synchronization Lost Interrupt Enable ClearRead indicates enabled status 0 = disabled 1 = enabledWriting 1 will clear interrupt enableWriting 0 has no effect
1	Reserved	R	0h	RESERVED

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 12-14. IRQENABLE_CLEAR Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	done_en_clr	R	0h	Raster or LIDD Frame Done (shared, depends on whether Raster or LIDD mode enabled) Interrupt Enable ClearRead indicates enabled status 0 = disabled 1 = enabledWriting 1 will clear interrupt enableWriting 0 has no effect

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.3 IRQEOI_VECTOR Register (offset = 68h) [reset = 0h]

IRQEOI_VECTOR is shown in Figure 12-28 and described in Table 12-15.

Figure 12-28. IRQEOI_VECTOR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
eoi_vector																															
R-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-15. IRQEOI_VECTOR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	eoi_vector	R	0h	Register Used by software. A write to this register informs the IPGeneric that the software has completed its Interrupt Service Routine. If the hardware has posted any pending interrupts that the software has not cleared or serviced, the IPGeneric s pulse interrupt line will be triggered again. The write data must be 0x0.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.4 CLKC_ENABLE Register (offset = 6Ch) [reset = 0h]

CLKC_ENABLE is shown in Figure 12-29 and described in Table 12-16.

Figure 12-29. CLKC_ENABLE Register

31	30	29	28	27	26	25	24
Reserved							
R-0h							

23	22	21	20	19	18	17	16
Reserved							
R-0h							

15	14	13	12	11	10	9	8
Reserved							
R-0h							

7	6	5	4	3	2	1	0
Reserved					dma_clk_en	lidd_clk_en	core_clk_en
R-0h					R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-16. CLKC_ENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	Reserved	R	0h	RESERVED
2	dma_clk_en	R/W	0h	Software Clock Enable for the DMA submodule. The DMA submodule runs on the L3 Clock domain.
1	lidd_clk_en	R/W	0h	Software Clock Enable for the LIDD submodule (character displays). The LIDD submodule runs on the System Clock (lcd_clk) domain
0	core_clk_en	R/W	0h	Software Clock Enable for the Core, which encompasses the Raster Active Matrix and Passive Matrix logic. The Core runs on the System Clock (lcd_clk) domain.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.5 CLKC_RESET Register (offset = 70h) [reset = 0h]

CLKC_RESET is shown in Figure 12-30 and described in Table 12-17.

Figure 12-30. CLKC_RESET Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved				main_rst	dma_rst	lidd_rst	core_rst
				R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-17. CLKC_RESET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-4	Reserved			
3	main_rst	R/W	0h	Software Reset for the entire LCD module. 1 = Reset Enable 0 = Reset Disable
2	dma_rst	R/W	0h	Software Reset for the DMA submodule. 1 = Reset Enable 0 = Reset Disable
1	lidd_rst	R/W	0h	Software Reset for the LIDD submodule (character displays). 1 = Reset Enable 0 = Reset Disable
0	core_rst	R/W	0h	Software Reset for the Core, which encompasses the Raster Active Matrix and Passive Matrix logic. 1 = Reset Enable 0 = Reset Disable

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.6 LIDD_CTRL Register (offset = Ch) [reset = 0h]

LIDD_CTRL is shown in Figure 12-31 and described in Table 12-18.

Figure 12-31. LIDD_CTRL Register

31	30	29	28	27	26	25	24
Reserved							
R-0h							

23	22	21	20	19	18	17	16
Reserved							
R-0h							

15	14	13	12	11	10	9	8
Reserved						dma_cs0_cs1	lidd_dma_en
R-0h						R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
cs1_e1_pol	cs0_e0_pol	ws_dir_pol	rs_en_pol	alepol	lidd_mode_sel		
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h		

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-18. LIDD_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	Reserved	R	0h	RESERVED
9	dma_cs0_cs1	R/W	0h	CS0/CS1 Select for LIDD DMA writes. 0 = DMA writes to LIDD CS0 1 = DMA writes for LIDD CS1
8	lidd_dma_en	R/W	0h	LIDD DMA Enable 0 = Deactivate DMA control of LIDD interface. DMA control is released upon completion of transfer of the current frame of data (LIDD Frame Done) after this bit is cleared. The MPU has direct read/write access to the panel in this mode. 1 = Activate DMA to drive LIDD interface to support streaming data to smart panels. The MPU cannot access the panel directly in this mode.
7	cs1_e1_pol	R/W	0h	Chip Select 1/Enable 1 (Secondary) Polarity Control 0 = Do Not Invert Chip Select 1/Enable 1. Chip Select 1 is active low by default. Enable 1 is active high by default. 1 = Invert Chip Select 1/Enable 1.
6	cs0_e0_pol	R/W	0h	Chip Select 0/Enable 0 (Secondary) Polarity Control. 0 = Do Not Invert Chip Select 0/Enable 0. Chip Select 0 is active low by default. Enable 0 is active high by default. 1 = Invert Chip Select 0/Enable 0.
5	ws_dir_pol	R/W	0h	Write Strobe/Direction Polarity Control 0 = Do Not Invert Write Strobe/Direction. Write Strobe/Direction is active low/write low by default. 1 = Invert Write Strobe/Direction
4	rs_en_pol	R/W	0h	Read Strobe/Direction Polarity Control. 0 = Do Not Invert Read Strobe/Direction. Read Strobe/Direction is active low/write low by default. 1 = Invert Read Strobe/Direction
3	alepol	R/W	0h	Address Latch Enable (ALE) Polarity Control 0 = Do Not Invert ALE. ALE is active low by default. 1 = Invert

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 12-18. LIDD_CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
2-0	lidd_mode_sel	R/W	0h	LIDD Mode Select. Selects type of LCD display interface for the LIDD to drive: 000b = Sync MPU68 001b = Async MPU68 010b = Sync MPU80 011b = Async MPU80 100b = Hitachi (Async) 101b = N/A 110b = N/A 111b = N/A LIDD Mode Select defines the function of LCD external pins as follows: lidd_mode_sel_value PIN 000b 0001b 010b 011b 100b lcd_cp EN EN Rs Rs N/A lcd_lp Dir Dir Ws Ws Dir lcd_fp Ale Ale Ale Ale Ale lcd_ac Cs0 Cs0 Cs0 Cs0 E0 lcd_mclk Mclk Cs1 Mclk Cs1 E1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.7 LIDD_CS0_CONF Register (offset = 10h) [reset = 440044h]

LIDD_CS0_CONF is shown in Figure 12-32 and described in Table 12-19.

Figure 12-32. LIDD_CS0_CONF Register

31	30	29	28	27	26	25	24
w_su					w_strobe		
R/W-0h					R/W-2h		

23	22	21	20	19	18	17	16
w_strobe			w_hold				r_su
R/W-2h			R/W-2h				R/W-0h

15	14	13	12	11	10	9	8
r_su				r_strobe			
R/W-0h				R/W-1h			

7	6	5	4	3	2	1	0
r_strobe		r_hold				ta	
R/W-1h		R/W-1h				R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-19. LIDD_CS0_CONF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	w_su	R/W	0h	Write Strobe Set-Up cycles. When performing a write access, this field defines the number of memclk cycles that Data Bus/Pad Output Enable, , the Direction bit, and Chip Select 0 have to be ready before the Write Strobe is asserted. The minimum value is 0x0.
26-21	w_strobe	R/W	2h	Write Strobe Duration cycles. Field value defines the number of memclk cycles for which the Write Strobe is held active when performing a write access. The minimum value is 0x1.
20-17	w_hold	R/W	2h	Write Strobe Hold cycles. Field value defines the number of memclk cycles for which Data Bus/Pas Output Enable, ALE, the Direction bit, and Chip Select 0 are held after the Write Strobe is de-asserted when performing write access. The minimum value is 0x1.
16-12	r_su	R/W	0h	Read Strobe Set-Up cycles. When performing a read access, this field defines the number of memclk cycles that Data Bus/Pad Output Enable, , the Direction bit, and Chip Select 0 have to be ready before the Read Strobe is asserted.
11-6	r_strobe	R/W	1h	Read Strobe Duration cycles. Field value defines the number of memclk cycles for which the Read Strobe is held active when performing a read access. The minimum value is 0x1.
5-2	r_hold	R/W	1h	Read Strobe Hold cycles. Field value defines the number of memclk cycles for which Data Bus/Pad Output Enable, , the Direction bit, and Chip Select 0 are held after the Read Strobe is deasserted when performing a read access. The minimum value is 0x1.
1-0	ta	R/W	0h	Field value defines the number of memclk (ta+1) cycles between the end of one CS0 device access and the start of another CS0 device access unless the two accesses are both Reads. In this case, this delay is not incurred. The minimum value is 0x0.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.8 LIDD_CS0_ADDR Register (offset = 14h) [reset = 0h]

LIDD_CS0_ADDR is shown in Figure 12-33 and described in Table 12-20.

Figure 12-33. LIDD_CS0_ADDR Register

31	30	29	28	27	26	25	24
Reserved							
R-0h							

23	22	21	20	19	18	17	16
Reserved							
R-0h							

15	14	13	12	11	10	9	8
adr_indx							
R/W-0h							

7	6	5	4	3	2	1	0
adr_indx							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-20. LIDD_CS0_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	Reserved	R	0h	RESERVED
15-0	adr_indx	R/W	0h	The LCD Controller supports a shared Address/Data output bus. A write to this register would initiate a bus write transaction. A read from this register would initiate a bus read transaction. CPU reads and writes to this register are not permitted if the LIDD module is in DMA mode (cfg_lidd_dma_en = 1).If the LIDD is being used as a generic bus interface, writing to this register can store adr_indx to an external transparent latch holding a 16-bit address.If the LIDD is being used to interface with a character based LCD panel in configuration mode, reading and writing to this register can be used to access the command instruction area of the panel.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.9 reg29 Register (offset = E8h) [reset = A00h]

reg29 is shown in Figure 12-34 and described in Table 12-21.

Figure 12-34. reg29 Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved						misr_lidd_count	
						R/W-Ah	

15	14	13	12	11	10	9	8
misr_lidd_count							
R/W-Ah							

7	6	5	4	3	2	1	0
Reserved						misr_control	
R-0h						R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-21. reg29 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-18	Reserved			
17-8	misr_lidd_count	R/W	Ah	MISR Transaction CounterData is fed into the MISR at start and end boundaries. For LIDD transactions, there are no clear start and stop events. Midd_lidd_count determines the number of LIDD output bus transactions before triggering a MISR stop event.
7-2	Reserved	R	0h	RESERVED
1-0	misr_control	R/W	0h	MISR control. 00 = MISR Hold 01 = MISR Load Seed Register 10 = MISR Run Once. Capture based on the LFSR start event and end on the stop event 11 = MISR Run. Capture continuously based on the LFSR start event to the LFSR stop event. Then, restart at the next LFSR start event.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.10 reg30 Register (offset = ECh) [reset = 0h]

reg30 is shown in Figure 12-35 and described in Table 12-22.

Figure 12-35. reg30 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
misr_seed																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-22. reg30 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	misr_seed	R/W	0h	MISR seed value

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.11 LIDD_CS0_DATA Register (offset = 18h) [reset = 0h]

LIDD_CS0_DATA is shown in Figure 12-36 and described in Table 12-23.

Figure 12-36. LIDD_CS0_DATA Register

31	30	29	28	27	26	25	24
Reserved							
R-0h							

23	22	21	20	19	18	17	16
Reserved							
R-0h							

15	14	13	12	11	10	9	8
data							
R/W-0h							

7	6	5	4	3	2	1	0
data							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-23. LIDD_CS0_DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	Reserved	R	0h	RESERVED
15-0	data	R/W	0h	The LCD Controller supports a shared Address/Data output bus. A write to this register would initiate a bus write transaction. A read from this register would initiate a bus read transaction. CPU reads and writes to this register are not permitted if the LIDD module is in DMA mode (cfg_lidd_dma_en = 1).If the LIDD is being used as a generic bus interface, writing to this register can store adr_indx to an external transparent latch holding a 16-bit address.If the LIDD is being used to interface with a character based LCD panel in configuration mode, reading and writing to this register can be used to access the command instruction area of the panel.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.12 reg31 Register (offset = F0h) [reset = 0h]

reg31 is shown in Figure 12-37 and described in Table 12-24.

Figure 12-37. reg31 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
misr_result																															
R-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-24. reg31 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	misr_result	R	0h	MISR result value

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.13 LIDD_CS1_CONF Register (offset = 1Ch) [reset = 420044h]

LIDD_CS1_CONF is shown in Figure 12-38 and described in Table 12-25.

Figure 12-38. LIDD_CS1_CONF Register

31	30	29	28	27	26	25	24
w_su					w_strobe		
R/W-0h					R/W-2h		

23	22	21	20	19	18	17	16
w_strobe			w_hold				r_su
R/W-2h			R/W-1h				R/W-0h

15	14	13	12	11	10	9	8
r_su				r_strobe			
R/W-0h				R/W-1h			

7	6	5	4	3	2	1	0
r_strobe		r_hold				ta	
R/W-1h		R/W-1h				R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-25. LIDD_CS1_CONF Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	w_su	R/W	0h	Write Strobe Set-Up cycles. When performing a write access, this field defines the number of memclk cycles that Data Bus/Pad Output Enable, , the Direction bit, and Chip Select 1 have to be ready before the Write Strobe is asserted. The minimum value is 0x0.
26-21	w_strobe	R/W	2h	Write Strobe Duration cycles. Field value defines the number of memclk cycles for which the Write Strobe is held active when performing a write access. The minimum value is 0x1.
20-17	w_hold	R/W	1h	Write Strobe Hold cycles. Field value defines the number of memclk cycles for which Data Bus/Pad Output Enable, ALE, the Direction bit, and Chip Select 1 are held after the Write Strobe is deasserted when performing a write access. The minimum value is 0x1.
16-12	r_su	R/W	0h	Read Strobe Set-Up cycles. When performing a read access, this field defines the number of memclk cycles that Data Bus/Pad Output Enable, , the Direction bit, and Chip Select 1 have to be ready before the Read Strobe is asserted. The minimum value is 0x0.
11-6	r_strobe	R/W	1h	Read Strobe Duration cycles. Field value defines the number of memclk cycles for which the Read Strobe is held active when performing a read access. The minimum value is 0x1.
5-2	r_hold	R/W	1h	Read Strobe Hold cycles. Field value defines the number of memclk cycles for which Data Bus/Pad Output Enable, , the Direction bit, and Chip Select 1 are held after the Read Strobe is deasserted when performing a read access. The minimum value is 0x1.
1-0	ta	R/W	0h	Field value defines the number of memclk (ta+1) cycles between the end of one CS1 device access and the start of another CS1 device access unless the two accesses are both Reads. In this case, this delay is not incurred. The minimum value is 0x0.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.14 LIDD_CS1_ADDR Register (offset = 20h) [reset = 0h]

LIDD_CS1_ADDR is shown in Figure 12-39 and described in Table 12-26.

Figure 12-39. LIDD_CS1_ADDR Register

31	30	29	28	27	26	25	24
Reserved							
R-0h							

23	22	21	20	19	18	17	16
Reserved							
R-0h							

15	14	13	12	11	10	9	8
adr_indx							
R/W-0h							

7	6	5	4	3	2	1	0
adr_indx							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-26. LIDD_CS1_ADDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	Reserved	R	0h	RESERVED
15-0	adr_indx	R/W	0h	The LCD Controller supports a shared Address/Data output bus. A write to this register would initiate a bus write transaction. A read from this register would initiate a bus read transaction. CPU reads and writes to this register are not permitted if the LIDD module is in DMA mode (cfg_lidd_dma_en = 1).If the LIDD is being used as a generic bus interface, writing to this register can store adr_indx to an external transparent latch holding a 16-bit address.If the LIDD is being used to interface with a character based LCD panel in configuration mode, reading and writing to this register can be used to access the command instruction area of the panel.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.15 LIDD_CS1_DATA Register (offset = 24h) [reset = 0h]

LIDD_CS1_DATA is shown in Figure 12-40 and described in Table 12-27.

Figure 12-40. LIDD_CS1_DATA Register

31	30	29	28	27	26	25	24
Reserved							
R-0h							

23	22	21	20	19	18	17	16
Reserved							
R-0h							

15	14	13	12	11	10	9	8
data							
R/W-0h							

7	6	5	4	3	2	1	0
data							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-27. LIDD_CS1_DATA Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	Reserved	R	0h	RESERVED
15-0	data	R/W	0h	The LCD Controller supports a shared Address/Data output bus. A write to this register would initiate a bus write transaction. A read from this register would initiate a bus read transaction. CPU reads and writes to this register are not permitted if the LIDD module is in DMA mode (cfg_lidd_dma_en = 1).If the LIDD is being used as a generic bus interface, writing to this register can store adr_indx to an external transparent latch holding a 16-bit address.If the LIDD is being used to interface with a character based LCD panel in configuration mode, reading and writing to this register can be used to access the command instruction area of the panel.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.16 CTRL Register (offset = 4h) [reset = 0h]

CTRL is shown in Figure 12-41 and described in Table 12-28.

Figure 12-41. CTRL Register

31	30	29	28	27	26	25	24
Reserved							
R-0h							

23	22	21	20	19	18	17	16
Reserved							
R-0h							

15	14	13	12	11	10	9	8
clkdiv							
R/W-0h							

7	6	5	4	3	2	1	0
Reserved						auto_uflow_restart	modesel
						R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-28. CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	Reserved	R	0h	RESERVED
15-8	clkdiv	R/W	0h	Clock divisor. Value (from 0-255) used to specify the frequency of the pixel clock (in Raster Mode) or MCLK in (LIDD mode) based on the lcd_clk frequency. Pixel clock frequency can range from lcd_clk/2 to lcd_clk/255 (clkdiv=0 or clkdiv=1 are N/A). MCLK can vary from lcd_clk to lcd_clk/255 (clkdiv=0 or clkdiv=1 sets MCLK=lcd_clk).
7-2	Reserved			
1	auto_uflow_restart	R/W	0h	0 = On an underflow, the software has to restart the module 1 = On an underflow, the hardware will restart on the next frame
0	modesel	R/W	0h	LCD Mode select. 0 = LCD Controller in LIDD Mode 1 = LCD Controller in Raster Mode

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.17 RASTER_CTRL Register (offset = 28h) [reset = 0h]

RASTER_CTRL is shown in Figure 12-42 and described in Table 12-29.

Figure 12-42. RASTER_CTRL Register

31	30	29	28	27	26	25	24
Reserved					tft24unpacked	tft24	stn565
					-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
tftmap	nibmode	palmode		reqdly			
R/W-0h	R/W-0h	R/W-0h		R/W-0h			

15	14	13	12	11	10	9	8
reqdly				Reserved		nono8b	rdorder
		R/W-0h				R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
lcdtft	Reserved					lcdbw	lcden
R/W-0h						R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-29. RASTER_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	Reserved			
26	tft24unpacked		0h	24 bit Mode PackingOnly used when lcdtft=1 and tft24= 10 = 24 bit pixels are packed into 32 bit boundaries, which means 4 pixels are saved in every three wordsWord 0: and#60 pix1\[7:0\]>and#60 pixl0\[23:0\]>Word 1: and#60 pix2\[15:0\]>and#60 pix1\[23:8\]>Word 2: and#60 pix3\[23:0\]>and#60 pix2\[\]> 1 = 24 bit pixels are stored unpacked in with the uppermost byte unused Word 0: and#60 Unused\[7:0\]>and#60 pix0\[23:0\]>Word 1: and#60 Unused\[7:0\]>and#60 pix1\[23:0\]>Word 2: and#60 Unused\[7:0\]>and#60 pix2\[23:0\]>Word 3: and#60 Unused\[7:0\]>and#60 pix3\[23:0\]>
25	tft24	R/W	0h	24 bit mode 0 = off 1 = on (24 bit data in Active mode)
24	stn565	R/W	0h	12 () mode 0 = off 1 = on (16 bit data in frame buffer, but only 12 bits are dithered and sent out)
23	tftmap	R/W	0h	Alternate Signal Mapping 0 = output pixel data for 1, 2, 4, and 8 bpp modes will be right alinged on lcd_pins (11:0) 1 = output pixel data for 1, 2, 4, and 8 bpp will be converted to 5,6,5, format and use pins (15:0)R3 R2 R1 R0 R3 G3 G2 G1 G0 G3 G2 B3 B2 B1 B0 B3

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 12-29. RASTER_CTRL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
22	nibmode	R/W	0h	Nibble Mode 0 = Nibble mode is disabled 1 = Nibble mode is enabled
21-20	palmode	R/W	0h	Palette Loading Mode 00 = Palette and data loading, reset value 01 = Palette loading only 10 = Data loading only. Data loading only mode is allowed only 24bpp input. That is, cfg_lcdtft= 1 and cfg_tft24= 1 . The DMA is programmed to send only the video portion of the framebuffer and not the palette data prepended to a framebuffer.
19-12	reqdly	R/W	0h	Palette Loading Delay When loading the Palette from DDR, palette data is burst into the internal Palette SRAM from the Async FIFO. 1, 2, and 4 bit per pixel framebuffer encodings use a fixed 16-word entry palette residing above the video data. The 8 bit per pixel framebuffer encoding uses a 256-word entry palette residing above the video data. Likewise, 12, 16, and 24 bit per pixel framebuffer encodings also define a 256-word entry palette even though these encodings will not do a full bit-depth palette lookup. However, the 256-word palette entry must still be read from DDR as a framebuffer is fetched. Bursting in 256 words in sequential lcd_clk cycles may cause the Async FIFO to underflow depending on the SOC DDR burst bandwidth. This 8 bit reqdly parameter pauses reading of the Palette data from the Async FIFO between each burst of 16 words. The delay is in terms of lcd_clk (system clock) cycles. Value (0-255) used to specify the number of system clock cycles that should be paused between bursts of 16 word reads from the Async FIFO while loading the Palette SRAM, Programming reqdly=00h disables this pause when loading the Palette table.
11-10	Reserved			
9	nono8b	R/W	0h	Mono 8 bit 0 = lcd_pixel_o\[3:0\] is used to output four pixel values to the panel each pixel clock transition 1 = lcd_pixel_o\[7:0\] is used to output eight pixel values to the panel each pixel clock transition.This bit is ignored in all other modes
8	rdorder	R/W	0h	Raster Data Order Select (this feature was referred to as Big Endian in the OMAP LCD spec, but in reality this feature is uniquely defined for the design and is not Endianism in the classic sense) 0 = Frame buffer data used to index into the Palette lookup is ordered from least-to-most significant bit/nibble/byte/halfword. 1 = Frame buffer data used to index into the Palette lookup is ordered from most-to-least significant bit/nibble/byte/halfword
7	lcdtft	R/W	0h	LCD 0 = Passive or display operation enabled, dither logic is enabled 1 = Active or display operation enabled, external palette and DAC required, dither logic bypassed, pin timing changes to support continuous pixel clock, output enable, vsync, and hsync.
6-2	Reserved			
1	lcdbw	R/W	0h	Only Applies for Passive Matrix Panels LCD Monochrome 0 = Color operation enabled 1 = Monochrome operation enabled
0	lcden	R/W	0h	LCD Controller Enable 0 = LCD controller disabled 1 = LCD controller enabled

DRAFT

Copyright © 2011, Texas Instruments Incorporated

12.5.18 RASTER_TIMING_0 Register (offset = 2Ch) [reset = 0h]

RASTER_TIMING_0 is shown in Figure 12-43 and described in Table 12-30.

Figure 12-43. RASTER_TIMING_0 Register

31	30	29	28	27	26	25	24
hbp							
R/W-0h							

23	22	21	20	19	18	17	16
hfp							
R/W-0h							

15	14	13	12	11	10	9	8
hsw						ppllsb	
R/W-0h						R/W-0h	

7	6	5	4	3	2	1	0
ppllsb				pplmsb	Reserved		
R/W-0h				R/W-0h	R-0h		

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-30. RASTER_TIMING_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	hbp	R/W	0h	Horizontal Back PorchEncoded value (from 1-256) used to specify the number of pixel clock periods to add to the beginning of a line transmission before the first set of pixels is output to the display (programmed value minus 1). Note that pixel clock is held in its inactive state during the beginning of the line wait period in passive display mode, and is permitted to transition in active display mode.
23-16	hfp	R/W	0h	Horizontal Front Porch Encoded value (from 1-256) used to specify the number of pixel clock periods to add to the end of a line transmission before line clock is asserted (programmed value minus 1). Note that pixel clock is held in its inactive state during the end of line wait period in passive display mode, and is permitted to transition in active display mode.
15-10	hsw	R/W	0h	Horizontal Sync Pulse WidthEncoded value (from 1-64) used to specify the number of pixel clock periods to pulse the line clock at the end of each line (programmed value minus 1). Note that pixel clock is held in its inactive state during the generation of line clock in passive display mode, and is permitted to transition in active display mode.
9-4	ppllsb	R/W	0h	Pixels-per-line LSB\[9:4\]Encoded LSB value (from 1-1024) used to specify the number of pixels contained within each line on the LCD display (programmed to value minus one). = 11'b{pplmsb, ppllsb, 4'b1111} + 1ex: pplmsb=1'b1, pppllsb=6' 0001000 = 11'b{1, 000100, 1111}+ 1 = 1104d (decimal) pixels per line
3	pplmsb	R/W	0h	Pixels-per-line \[10\]Needed in order to support up to 2048 ppl.
2-0	Reserved	R	0h	RESERVED

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.19 RASTER_TIMING_1 Register (offset = 30h) [reset = 0h]

RASTER_TIMING_1 is shown in Figure 12-44 and described in Table 12-31.

Figure 12-44. RASTER_TIMING_1 Register

31	30	29	28	27	26	25	24
vbp							
R/W-0h							

23	22	21	20	19	18	17	16
vfp							
R/W-0h							

15	14	13	12	11	10	9	8
vsw						lpp	
R/W-0h						R/W-0h	

7	6	5	4	3	2	1	0
lpp							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-31. RASTER_TIMING_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	vbp	R/W	0h	Vertical Back PorchValue (from 0-255) use to specify the number of line clock periods to add to the beginning of a frame before the first set of pixels is output to the display. Note that line clock transitions during the insertion of the extra line clock periods.
23-16	vfp	R/W	0h	Vertical Front Porch Value (from 0-255) used to specify the number of line clock periods to add to the end of each frame. Note that the line clock transitions during the insertion of the extra line clock periods.
15-10	vsw	R/W	0h	Vertical Sync Width PulseIn active mode (lcdtft=1), encoded value (from 1-64) used to specify the number of line clock periods to pulse the lcd_fp pin at the end of each frame after the end of the end of frame wait (vfp) period elapses. The frame clock is used as the VSYNC signal in active mode.In passive mode (lcdtft=0), encoded value (from 1-64) used to specify the number of extra line clock periods to insert after the vertical front porch (vfp) period has elapsed. Note that the width of lcd_fp is not affected by vsw in passive mode and line clock tranisitions during the insertion of the extra line clock periods (programmed to value minus one).
9-0	lpp	R/W	0h	Lines Per PanelEncoded value (from 1-2048) used to specify the number of lines per panel. It represents the total number of lines on the LCD (programmed to value minus one). Bit 10 of this field is in RASTER_TIMING_2.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.20 RASTER_TIMING_2 Register (offset = 34h) [reset = 0h]

RASTER_TIMING_2 is shown in Figure 12-45 and described in Table 12-32.

Figure 12-45. RASTER_TIMING_2 Register

31	30	29	28	27	26	25	24
Reserved					lpp_b10	phsvs_on_off	phsvs_rf
					R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
ieo	ipc	ihs	ivs	acbi			
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h			

15	14	13	12	11	10	9	8
acb							
R/W-0h							

7	6	5	4	3	2	1	0
Reserved							
R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-32. RASTER_TIMING_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-27	Reserved			
26	lpp_b10	R/W	0h	Lines Per Panel Bit 10Bit 10 of the lpp field in RASTER_TIMING_1.
25	phsvs_on_off	R/W	0h	Hsync/Vsync Pixel Clock Control On/Off 0 = lcd_lp and lcd_fp are driven on opposite edges of pixel clock than the lcd_pixel_o 1 = lcd_lp and lcd_fp are driven according to bit 24
24	phsvs_rf	R/W	0h	Program HSYNC/VSYNC Rise or Fall 0 = lcd_lp and lcd_fp are driven on the falling edge of pixel clock (bit 25 must be set to 1). 1 = lcd_lp and lcd_fpare driven on the rising edge of pixel clock (bit 25 must be set to 1).
23	ieo	R/W	0h	Invert Output Enable 0 = lcd_ac pin is active high in active display mode 1 = lcd_ac pin is active low in active display modeActive display mode: data driven out of the LCD's data lines on programmed pixel clock edge where AC-bias is active. Note that ieo is ignored in passive display mode.
22	ipc	R/W	0h	Invert Pixel Clock 0 = Data is driven on the LCD's data lines on the rising edge of lcd_cp 1 = Data is driven on the LCD's data lines in the falling edge of lcd_cp
21	ihs	R/W	0h	Invert Hsync 0 = lcd_lp pin is active high and inactive low 1 = lcd_lp pin is active low and inactive highActive and passive mode: horizontal sync pulse/line clock active between lines, after the end of line wait period
20	ivs	R/W	0h	Invert Vsync 0 = lcd_fp pin is active high and inactive low 1 = lcd_fp pin is active low and inactive highActive mode: vertical sync pulse active between frames, after end of frame wait periodPassive mode: frame clock active during first line of each frame
19-16	acbi	R/W	0h	AC Bias Pins Transitions per InterruptValue (from 0 to 15) used to specify the number of AC Bias pin transitions to count before setting the line count status (lcs) bit, signaling an interrupt request. Counter frozen when lcd is set, and is restarted when lcs is cleared by software. This function is disabled when acbi = b'0000'.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 12-32. RASTER_TIMING_2 Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
15-8	acb	R/W	0h	AC Bias Pin FrequencyValue (from 0-255) used to specify the number of line clocks to count before transitioning the AC Bias pin. This pin is used to periodically invert the polarity of the power supply to prevent DC charge build-up within the display. acb = Number of line clocks/toggle of the lcd_ac pin.
7-0	Reserved	R	0h	RESERVED

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.21 RASTER_SUBPANEL Register (offset = 38h) [reset = 0h]

RASTER_SUBPANEL is shown in Figure 12-46 and described in Table 12-33.

Figure 12-46. RASTER_SUBPANEL Register

31	30	29	28	27	26	25	24
spen	Reserved	hols	Reserved			lppt	
R/W-0h	R-0h	R/W-0h				R/W-0h	

23	22	21	20	19	18	17	16
lppt							
R/W-0h							

15	14	13	12	11	10	9	8
dpdlsb							
R/W-0h							

7	6	5	4	3	2	1	0
dpdlsb							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-33. RASTER_SUBPANEL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	spen	R/W	0h	Sub Panel Enable 0 = function disabled 1 = sub-panel function mode enabled
30	Reserved	R	0h	RESERVED
29	hols	R/W	0h	High or Low SignalThis field indicates the position of the sub-panel compared to the LPPT value
28-26	Reserved			
25-16	lppt	R/W	0h	Line Per Panel ThresholdThis field defines the number of lines to be refreshed in the range {1:2048}. Bit10 of this field is in RASTER_SUBPANEL2.
15-0	dpdlsb	R/W	0h	Default Pixel Data LSB\[15:0\]DPD defines the default value of the pixel data sent to the panel for the lines until LPPT is reach or after passing LPPT

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.22 RASTER_SUBPANEL2 Register (offset = 3Ch) [reset = 0h]

RASTER_SUBPANEL2 is shown in Figure 12-47 and described in Table 12-34.

Figure 12-47. RASTER_SUBPANEL2 Register

31	30	29	28	27	26	25	24
Reserved							
R-0h							

23	22	21	20	19	18	17	16
Reserved							
R-0h							

15	14	13	12	11	10	9	8
Reserved							lppt_b10
R-0h							R/W-0h

7	6	5	4	3	2	1	0
dpdmsb							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-34. RASTER_SUBPANEL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R	0h	RESERVED
8	lppt_b10	R/W	0h	Lines Per Panel Threshold Bit 10This register is Bit 10 of the lppt field in RASTER_SUBPANEL.
7-0	dpdmsb	R/W	0h	Default Pixel Data \[\]DPD defines the default value of the pixel data sent to the panel for the lines until LPPT is reached or after passing the LPPT

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.23 LCDDMA_CTRL Register (offset = 40h) [reset = 0h]

LCDDMA_CTRL is shown in Figure 12-48 and described in Table 12-35.

Figure 12-48. LCDDMA_CTRL Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved					dma_master_prio		
						R/W-0h	

15	14	13	12	11	10	9	8
Reserved					th_fifo_ready		
						R/W-0h	

7	6	5	4	3	2	1	0
Reserved	burst_size			byte_swap	Reserved	bigendian	frame_mode
		R/W-0h		R/W-0h	R-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 12-35. LCDDMA_CTRL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-19	Reserved			
18-16	dma_master_prio	R/W	0h	Priority for the L3 OCP Master Bus
15-11	Reserved			
10-8	th_fifo_ready	R/W	0h	DMA FIFO threshold. The DMA FIFO becomes ready when the number of words specified by this register from the frame buffer have been loaded. 000 = 8001 = 16010 = 32011 = 64100 = 128101 = 256110 = 512111 = reserved
7	Reserved			
6-4	burst_size	R/W	0h	Burst Size setting for DMA transfers (all DMA transfers are 32 bits wide): 000 = burst size of 1001 = burst size of 2010 = burst size of 4011 = burst size of 8100 = burst size of 16101 = N/A 110 = N/A 111 = N/A
3	byte_swap	R/W	0h	Swap the bytes within a half-word of the dma transferGiven a word: 01230 = No swap : 01231 = Swap : 1032
2	Reserved	R	0h	RESERVED
1	bigendian	R/W	0h	Big Endian Enable. Use this bit when the processor is operating in Big Endian mode AND writes to the frame buffer(s) are less than 32 bits wide. Only in this scenario do we need to change the byte alignment for data coming into the FIFO from the frame buffer(s). 0 = Big Endian data reordering disabled 1 = Big Endian data reordering enabled
0	frame_mode	R/W	0h	Frame Mode 0 = one frame buffer (FB0 only) used 1 = two frame buffers used,. DMA ping-pongs between FB0 and FB1 in this mode

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.24 LCDDMA_FB0_BASE Register (offset = 44h) [reset = 0h]

LCDDMA_FB0_BASE is shown in Figure 12-49 and described in Table 12-36.

Figure 12-49. LCDDMA_FB0_BASE Register

31	30	29	28	27	26	25	24
fb0_base							
R/W-0h							

23	22	21	20	19	18	17	16
fb0_base							
R/W-0h							

15	14	13	12	11	10	9	8
fb0_base							
R/W-0h							

7	6	5	4	3	2	1	0
fb0_base						Reserved	
R/W-0h						R-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-36. LCDDMA_FB0_BASE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	fb0_base	R/W	0h	Frame Buffer 0 Base Address pointer
1-0	Reserved	R	0h	RESERVED

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.25 LCDDMA_FB0_CEILING Register (offset = 48h) [reset = 0h]

LCDDMA_FB0_CEILING is shown in Figure 12-50 and described in Table 12-37.

Figure 12-50. LCDDMA_FB0_CEILING Register

31	30	29	28	27	26	25	24
fb0_ceil							
R/W-0h							

23	22	21	20	19	18	17	16
fb0_ceil							
R/W-0h							

15	14	13	12	11	10	9	8
fb0_ceil							
R/W-0h							

7	6	5	4	3	2	1	0
fb0_ceil						Reserved	
R/W-0h						R-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-37. LCDDMA_FB0_CEILING Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	fb0_ceil	R/W	0h	Frame Buffer 0 Ceiling Address pointer
1-0	Reserved	R	0h	RESERVED

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.26 LCDDMA_FB1_BASE Register (offset = 4Ch) [reset = 0h]

LCDDMA_FB1_BASE is shown in Figure 12-51 and described in Table 12-38.

Figure 12-51. LCDDMA_FB1_BASE Register

31	30	29	28	27	26	25	24
fb1_base							
R/W-0h							

23	22	21	20	19	18	17	16
fb1_base							
R/W-0h							

15	14	13	12	11	10	9	8
fb1_base							
R/W-0h							

7	6	5	4	3	2	1	0
fb1_base						Reserved	
R/W-0h						R-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-38. LCDDMA_FB1_BASE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	fb1_base	R/W	0h	Frame Buffer 1 Base Address pointer
1-0	Reserved	R	0h	RESERVED

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.27 LCDDMA_FB1_CEILING Register (offset = 50h) [reset = 0h]

LCDDMA_FB1_CEILING is shown in Figure 12-52 and described in Table 12-39.

Figure 12-52. LCDDMA_FB1_CEILING Register

31	30	29	28	27	26	25	24
fb1_ceil							
R/W-0h							

23	22	21	20	19	18	17	16
fb1_ceil							
R/W-0h							

15	14	13	12	11	10	9	8
fb1_ceil							
R/W-0h							

7	6	5	4	3	2	1	0
fb1_ceil						Reserved	
R/W-0h						R-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-39. LCDDMA_FB1_CEILING Register Field Descriptions

Bit	Field	Type	Reset	Description
31-2	fb1_ceil	R/W	0h	Frame Buffer 1 Ceiling Address pointer
1-0	Reserved	R	0h	RESERVED

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.28 SYSCONFIG Register (offset = 54h) [reset = 28h]

SYSCONFIG is shown in Figure 16-2 and described in Table 16-5.

Figure 12-53. SYSCONFIG Register

31	30	29	28	27	26	25	24
Reserved							
R-0h							

23	22	21	20	19	18	17	16
Reserved							
R-0h							

15	14	13	12	11	10	9	8
Reserved							
R-0h							

7	6	5	4	3	2	1	0
Reserved		standbymode		idlemode		Reserved	
R-0h		R/W-2h		R/W-2h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-40. SYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-6	Reserved	R	0h	RESERVED
5-4	standbymode	R/W	2h	Configuration of the local initiator state management mode.By definition, initiator may generate read/write transaction as long as it is out of STANDBY state 0x0 : Force-standby mode: local initiator is unconditionally placed in standby state.Backup mode, for debug only 0x1 : No-standby mode: local initiator is unconditionally placed out of standby state.Backup mode, for debug only 0x2 : Smart-standby mode: local initiator standby status depends on local conditions, i.e. the module's functional requirement from the initiator.IP module shall not generate (initiator-related) wakeup events 0x3 : Smart-Standby wakeup-capable mode: local initiator standby status depends on local conditions, i.e. the module's functional requirement from the initiator. IP module may generate (master-related) wakeup events when in standby state.Mode is only relevant if the appropriate IP module "mwakeup" output is implemented
3-2	idlemode	R/W	2h	Configuration of the local target state management mode.By definition, target can handle read/write transaction as long as it is out of IDLE state 0x0 : Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, i.e. regardless of the IP module's internal requirements.Backup mode, for debug only 0x1 : No-idle mode: local target never enters idle state.Backup mode, for debug only 0x2 : Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements.IP module shall not generate (IRQ- or DMA-request-related) wakeup events 0x3 : Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements.IP module may generate (IRQ- or DMA-request-related) wakeup events when in idle state.Mode is only relevant if the appropriate IP module "swakeup" output(s) is (are) implemented
1-0	Reserved			

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.29 IRQSTATUS_RAW Register (offset = 58h) [reset = 0h]

IRQSTATUS_RAW is shown in Figure 12-54 and described in Table 12-41.

Figure 12-54. IRQSTATUS_RAW Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved						eof1_raw_set	eof0_raw_set
						R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
Reserved	pl_raw_set	fuf_raw_set	Reserved	acb_raw_set	sync_raw_set	Reserved	done_raw_set
	R/W-0h	R/W-0h	R-0h	R/W-0h	R-0h		R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-41. IRQSTATUS_RAW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	Reserved			
9	eof1_raw_set	R/W	0h	DMA End-of-Frame 1 Raw Interrupt Status and SetRead indicates raw status 0 = inactive 1 = activeWriting 1 will set statusWriting 0 has no effect
8	eof0_raw_set	R/W	0h	DMA End-of-Frame 0 Raw Interrupt Status and SetRead indicates raw status 0 = inactive 1 = activeWriting 1 will set statusWriting 0 has no effect
7	Reserved			
6	pl_raw_set	R/W	0h	DMA Palette Loaded Raw Interrupt Status and SetRead indicates raw status 0 = inactive 1 = activeWriting 1 will set statusWriting 0 has no effect
5	fuf_raw_set	R/W	0h	DMA FIFO Underflow Raw Interrupt Status and SetLCD dithering logic not supplying data to FIFO at a sufficient rate, FIFO has completely emptied and data pin driver logic has attempted to take added data from FIFO.Read indicates raw status 0 = inactive 1 = activeWriting 1 will set statusWriting 0 has no effect
4	Reserved	R	0h	RESERVED
3	acb_raw_set	R/W	0h	For Passive Matrix Panels OnlyAC Bias Count Raw Interrupt Status and SetAC bias transition counter has decremented to zero, indicating that the lcd_ac_o line has transitioned the number of times which is specified by the acbi control bit-field. The counter is reloaded with the value in acbi but it is disabled until the user clears ABC.Read indicates raw status 0 = inactive 1 = activeWriting 1 will set statusWriting 0 has no effect
2	sync_raw_set	R	0h	Frame Synchronization Lost Raw Interrupt Status and Set Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect
1	Reserved			
0	done_raw_set	R	0h	Raster or LIDD Frame Done (shared, depends on whether Raster or LIDD mode enabled) Raw Interrupt Status and SetRead indicates raw status 0 = inactive 1 = activeWriting 1 will set statusWriting 0 has no effect

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.30 IRQSTATUS Register (offset = 5Ch) [reset = 0h]

IRQSTATUS is shown in Figure 12-55 and described in Table 12-42.

Figure 12-55. IRQSTATUS Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved						eof1_en_clr	eof0_en_clr
						R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
Reserved	pl_en_clr	fuf_en_clr	Reserved	acb_en_clr	sync_en_clr	Reserved	done_en_clr
	R/W-0h	R/W-0h		R/W-0h	R/W-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-42. IRQSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	Reserved			
9	eof1_en_clr	R/W	0h	DMA End-of-Frame 1 Enabled Interrupt and ClearRead indicates enabled (masked) status 0 = inactive 1 = activeWriting 1 will clear interrupt enableWriting 0 has no effect
8	eof0_en_clr	R/W	0h	DMA End-of-Frame 0 Enabled Interrupt and ClearRead indicates enabled (masked) status 0 = inactive 1 = activeWriting 1 will clear interrupt enableWriting 0 has no effect
7	Reserved			
6	pl_en_clr	R/W	0h	DMA Palette Loaded Enabled Interrupt and ClearRead indicates enabled (masked) status 0 = inactive 1 = activeWriting 1 will clear interrupt enableWriting 0 has no effect
5	fuf_en_clr	R/W	0h	DMA FIFO Underflow Enabled Interrupt and ClearLCD dithering logic not supplying data to FIFO at a sufficient rate, FIFO has completely emptied and data pin driver logic has attempted to take added data from FIFO.Read indicates enabled (masked) status 0 = inactive 1 = activeWriting 1 will clear interrupt enableWriting 0 has no effect
4	Reserved			
3	acb_en_clr	R/W	0h	For Passive Matrix Panels OnlyAC Bias Count Enabled Interrupt and ClearAC bias transition counter has decremented to zero, indicating that the lcd_ac_o line has transitioned the number of times which is specified by the acbi control bit-field. The counter is reloaded with the value in acbi but it is disabled until the user clears ABC.Read indicates enabled (masked) status 0 = inactive 1 = activeWriting 1 will clear interrupt enableWriting 0 has no effect
2	sync_en_clr	R/W	0h	Frame Synchronization Lost Enabled Interrupt and ClearRead indicates enabled status 0 = inactive 1 = activeWriting 1 will clear interrupt enableWriting 0 has no effect
1	Reserved	R	0h	RESERVED
0	done_en_clr	R	0h	Raster or LIDD Frame Done (shared, depends on whether Raster or LIDD mode enabled) Enabled Interrupt and ClearRead indicates enabled status 0 = inactive 1 = activeWriting 1 will clear interrupt enableWriting 0 has no effect

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

12.5.31 IRQENABLE_SET Register (offset = 60h) [reset = 0h]

IRQENABLE_SET is shown in Figure 12-56 and described in Table 12-43.

Figure 12-56. IRQENABLE_SET Register

31	30	29	28	27	26	25	24
Reserved							
-0h							

23	22	21	20	19	18	17	16
Reserved							
-0h							

15	14	13	12	11	10	9	8
Reserved						eof1_en_set	eof0_en_set
-0h						R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
Reserved	pl_en_set	fuf_en_set	Reserved	acb_en_set	sync_en_set	Reserved	done_en_set
	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W-0h		R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 12-43. IRQENABLE_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	Reserved		0h	
9	eof1_en_set	R/W	0h	DMA End-of-Frame 1 Interrupt Enable SetRead indicates enabled (mask) status 0 = disabled 1 = enabledWriting 1 will set interrupt enableWriting 0 has no effect
8	eof0_en_set	R/W	0h	DMA End-of-Frame 0 Interrupt Enable Set Read indicates enabled (mask) status 0 = disabled 1 = enabled Writing 1 will set interrupt enable Writing 0 has no effect
7	Reserved			
6	pl_en_set	R/W	0h	DMA Palette Loaded Interrupt Enable SetRead indicates enabled (mask) status 0 = disabled 1 = enabledWriting 1 will set interrupt enableWriting 0 has no effect
5	fuf_en_set	R/W	0h	DMA FIFO Underflow Interrupt Enable SetLCD dithering logic not supplying data to FIFO at a sufficient rate, FIFO has completely emptied and data pin driver logic has attempted to take added data from FIFO.Read indicates enabled (mask) status 0 = disabled 1 = enabledWriting 1 will set interrupt enableWriting 0 has no effect
4	Reserved	R	0h	RESERVED
3	acb_en_set	R/W	0h	For Passive Matrix Panels Only AC Bias Count Interrupt Enable Set AC bias transition counter has decremented to zero, indicating that the lcd_ac_o line has transitioned the number of times which is specified by the acbi control bit-field. The counter is reloaded with the value in acbi but it is disabled until the user clears ABC. Read indicates enabled (mask) status 0 = disabled 1 = enabled Writing 1 will set interrupt enable Writing 0 has no effect
2	sync_en_set	R/W	0h	Frame Synchronization Lost Interrupt Enable SetRead indicates enabled (mask) status 0 = disabled 1 = enabledWriting 1 will set interrupt enableWriting 0 has no effect
1	Reserved			

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 12-43. IRQENABLE_SET Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	done_en_set	R	0h	Raster or LIDD Frame Done (shared, depends on whether Raster or LIDD mode enabled) Interrupt Enable SetRead indicates enabled (mask) status 0 = disabled 1 = enabledWriting 1 will set interrupt enableWriting 0 has no effect

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Chapter 13

SPRUH73–July 2011

Ethernet Subsystem

This chapter describes the ethernet subsystem of the device.

Topic **Page**

13.1 Introduction **1578**
13.2 Integration **1580**
13.3 Functional Description **1583**
13.4 Software Operation **1625**
13.5 CPSW_ALE Registers **1629**
13.6 CPSW_CPDMA Registers **1644**
13.7 CPSW_CPTS Registers **1729**
13.8 CPSW_PORT Registers **1742**
13.9 CPSW_SL Registers **1798**
13.10 CPSW_SS Registers **1811**
13.11 CPSW_WR Registers **1825**

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.1 Introduction

Described in the following sections is the 3-port switch (3PSW) Ethernet subsystem. The 3-port switch gigabit ethernet subsystem provides ethernet packet communication and can be configured as an ethernet switch. It provides the gigabit media independent interface (GMII),reduced gigabit media independent interface (RGMII), reduced media independent interface (RMII), the management data input output (MDIO) for physical layer device (PHY) management.

13.1.1 Features

The general features of the ethernet switch subsystem are:

- Two 10/100/1000 Ethernet ports with GMII, RMII and RGMII interfaces
- Wire rate switching (802.1d)
- Non Blocking switch fabric
- Flexible logical FIFO based packet buffer structure
- Four priority level QOS support (802.1p)
- CPPI 3.1 compliant DMA controllers
- Support for Audio/Video Bridging (P802.1Qav/D6.0)
- Support for IEEE 1588 Clock Synchronization (2008 Annex D and Annex F)
  - Timing FIFO and time stamping logic inside the SS
- Device Level Ring (DLR) Support
- Address Lookup Engine
  - 1024 addresses plus VLANs
  - Wire rate lookup
  - VLAN support
  - Host controlled time-based aging
  - Spanning tree support
  - L2 address lock and L2 filtering support
  - MAC authentication (802.1x)
  - Receive or destination based Multicast and Broadcast limits
  - MAC address blocking
  - Source port locking
  - OUI host accept/deny feature
- Flow Control Support (802.3x)
- EtherStats and 802.3Stats RMON statistics gathering (shared)
- Support for external packet dropping engine
- CPGMAC_SL transmit to CPGMAC_SL receive Loopback mode (digital loopback) supported
- CPGMAC_SL receive to CPGMAC_SL transmit Loobback mode (FIFO loopback) supported
- Maximum frame size 2016 bytes (2020 with VLAN)
- 8k (2048 x 32) internal CPPI buffer descriptor memory
- MDIO module for PHY Management
- Programmable interrupt control with selected interrupt pacing
- Emulation Support.
- Programmable transmit Inter-Packet Gap (IPG)
- Reset isolation

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.1.2 Unsupported Features

There are 18 level interrupts from the CPGMAC module and 2 (used) level interrupts from the MDIO module. The CPSW_3GSS includes an interrupt combiner/pacer to combine these interrupts together to produce 4 interrupt outputs (per processing core). This device does not split processing among multiple cores but allows servicing of the Core0 interrupts by the Cortex A8 or the PRUSS.

The unsupported CPGMAC features in the device are shown in the following table.

Table 13-1. Unsupported CPGMAC Features

Feature	Reason
Multi-core split processing	Core 1 and Core 2 interrupts not connected.
GMII	Only 4 Rx/Tx data pins are pinned out for each port. The device supports MII (on GMII interface), RGMII, and RMII interfaces only
Phy link status	The MLINK inputs are nott pinned out. Phy link status outputs can be connected to device GPIOs.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.2 Integration

This device includes a single instantiation of the three-port Gigabit Ethernet Switch Subsystem (CPSW_3GSS_RG). The switch provides 2 external ethernet ports (ports 1 and 2) and an internal CPPI interface port (port 0) with AVB/Industrial Ethernet and 802.1ae support. The subsystem consists of:

- One instance of the 3-port Gigabit switch CPSW-3G, which contains:
  - 2 CPGMAC_SL 10/100/1000 ethernet port modules with GMII interface
- Two RGMII interface modules
- Two RMII interface modules
- One MDIO interface module
- One Interrupt Controller module
- Local CPPI memory of size 8K Bytes

The integration of the Ethernet Switch is shown in Figure 13-1

Figure 13-1. Ethernet Switch Integration

13.2.1 Ethernet Switch Connectivity Attributes

The general connectivity attributes for the Ethernet Switch module are shown in Table 13-2.

Table 13-2. Ethernet Switch Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_CPSW_125MHZ_GCLK (Main) PD_PER_CPSW_250MHZ_GCLK (MHZ_250_CLK) PD_PER_CPSW_50MHZ_GCLK (MHZ_50_CLK) PD_PER_CPSW_5MHZ_GCLK (MHZ_5_CLK) PD_PER_CPSW_CPTS_RFT_CLK (CPTS_RFT_CLK)
Reset Signals	CPSW_MAIN_ARST_N CPSW_ISO_MAIN_ARST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	4 Interrupts RX_THRESH – Receive Threshold interrupt (nonpaced) RX – Receive interrupt (paced) TX – Transmit interrupt (paced) Misc – Other interrupts All Ethernet Switch interrupts go to MPU Subsystem and PRUSS. The Subsystem contains 3 sets of interrupts C0, C1, and C2 to allow for split core processing of packets. On this device, only the C0 version of the interrupts is used.
DMA Requests	None
Physical Address	L4 Fast slave port L3 Fast initiator port

13.2.2 Ethernet Switch Clock and Reset Management

The ethernet switch controller operates in it's own clock domain and its initiator and target interfaces are connected to the L3/L4 through asynchronous bridges. The OCP interfaces are driven by the MAIN clock input. Additional reference clock inputs are provided for operating the various ethernet ports at different rates.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 13-3. Ethernet Switch Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
rft_clk Gigabit GMII Tx Reference clock	125 MHz	Tied low	not supported
main_clk Logic/Interface clock	125 MHz	CORE_CLKOUTM5 / 2	pd_per_cpsw_125mhz_gclk from PRCM
mhz250_clk Gigabit RGMII Reference clock	250 MHz	CORE_CLKOUTM5	pd_per_cpsw_250mhz_gclk from PRCM
mhz50_clk RMII and 100mbps RGMII Reference clock	50 MHz	CORE_CLKOUTM5 / 5	pd_per_cpsw_50mhz_gclk from PRCM
mhz5_clk 10 mbpsRGMII Reference clock	5 MHz	CORE_CLKOUTM5 / 50	pd_per_cpsw_5mhz_gclk from PRCM
cpts_rft_clk IEEE 1588 clock	250 MHz	CORE_CLKOUTM4 CORE_CLKOUTM5	pd_per_cpsw_cpts_rft_clk from PRCM
gmii1_mr_clk GMII Port 1 Receive clock	25 MHz	External Pin	gmii1_rxclk_in from GMII1_RCLK pad
gmii2_mr_clk GMII Port 2 Receive clock	25 MHz	External Pin	gmii2_rxclk_in from GMII2_RCLK pad
gmii1_mt_clk GMII Port 1 Transmit clock	25 MHz	External Pin	gmii1_txclk_in from GMII1_TCLK pad
gmii2_mt_clk GMII Port 2 Transmit clock	25 MHz	External Pin	gmii2_txclk_in from GMII2_TCLK pad
rgmii1_rxc_clk RGMII Port 1 Receive clock	250 MHz	External Pin	rgmii1_rclk_in from RGMII1_RCLK pad
rgmii2_rxc_clk RGMII Port 2 Receive clock	250 MHz	External Pin	rgmii2_rclk_in from RGMII2_RCLK pad
rmii1_mhz_50_clk RMII Port 1 Reference clock	50 MHz	External Pin	rmii1_refclk_in from RMII1_REFCLK pad
rmii2_mhz_50_clk RMII Port 2 Reference clock	50 MHz	External Pin	rmii2_refclk_in from RMII2_REFCLK pad

13.2.3 Ethernet Switch Pin List

The external signals for the Ethernet Switch module are shown in the following table.

Table 13-4. Ethernet Switch Pin List

Pin	Type*	Description
GMIIx_RXCLK	I	GMII/MII Receive clock
GMIIx_RXD[7:0]	I	GMII/MII Receive data
GMIIx_RXDV	I	GMII/MII Receive data valid
GMIIx_RXER	I	GMII/MII Receive error
GMIIx_COL	I	GMII/MII Collision detect
GMIIx_CRS	I	GMII/MII Carrier sense
GMIIx_GTXCLK	O	GMII Source synchronous transmit clock
GMIIx_TXCLK	I	GMII/MII Transmit clock
GMII_TXD[7:0]	O	GMII/MII Transmit data
GMIIx_TXEN	O	GMII/MII Transmit enable
RGMIIx_RCLK	I	RGMII Receive clock
RGMIIx_RCTL	I	RGMII Receive control
RGMIIx_RD[3:0]	I	RGMII Receive data
RGMIIx_TCLK	O	RGMII Transmit clock
RGMIIx_TCTL	O	RGMII Transmit control

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 13-4. Ethernet Switch Pin List (continued)

Pin	Type*	Description
RGMIIx_TD[3:0]	O	RGMII Transmit data
RMIIx_RXD[1”0]	I	RMII Receiver data
RMIIx_RXERR	I	RMII Receiver error
RMIIx_CRS_DV	I	RMII Carrier sense / Data valid
RMIIx_TXEN	O	RMII Transmit enable
RMIIx_TXD[1:0]	O	RMII Transmit data
MDIO_MCLK	O	MDIO Serial clock
MDIO_MDIO	I/O	MDIO Serial data

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.3 Functional Description

The 3 port switch (3PSW) Ethernet Subsystem peripheral are compliant to the IEEE Std 802.3 Specification. The 3PSW Ethernet Subsystem contains two RGMII/RMII interfaces,one CPPI 3.0 interface, Interrupt Controller, MDI0 and CPSW_3G which contains two GMII interfaces as shwon in the figure1 3PSW Ethernet Subsytem Top Level Block Diagram.

The Subsystem Modules are discussed in detail in the following sections.

13.3.1 CPSW_3G

The CPSW_3G GMII interfaces are compliant to the IEEE Std 802.3 Specification.

The CPSW_3G contains two CPGMAC_SL interfaces (ports 1 and 2), one CPPI 3.0 interface Host Port (port 0), Common Platform Time Sync (CPTS), ALE Engine and CPDMA.

The top level block diagram of CPSW_3G is shown below:

Figure 13-2. CPSW_3G Block Diagram

13.3.1.1 CPDMA RX and TX Interfaces

The CPDMA submodule is a CPPI 3.0 compliant packet DMA transfer controller. The CPPI 3.0 interface is port 0.

After reset, initialization, and configuration the host may initiate transmit operations. Transmit operations are initiated by host writes to the appropriate transmit channel head descriptor pointer contained in the STATERAM block. The transmit DMA controller then fetches the first packet in the packet chain from memory in accordance with CPPI 3.0 protocol. The DMA controller writes the packet into the external transmit FIFO in 64-byte bursts (maximum).

Receive operations are initiated by host writes to the appropriate receive channel head descriptor pointer after host initialization and configuration. The receive DMA controller writes the receive packet data to external memory in accordance with CPPI 3.0 protocol. See the **CPPI Buffer Descriptors** section for detailed description of Buffer Descriptors

13.3.1.1.1 Receive DMA Interface

The Receive DMA is an eight channel CPPI 3.0 compliant interface. Each channel has a single queue for frame reception.

13.3.1.1.1.1 Receive DMA Host Configuration

To configure the Rx DMA for operation the host must perform the following:

- Initialize the receive addresses.
- Initialize the **Rx_HDP** Registers to zero.
- Enable the desired receive interrupts in the **IntMask** register.
- Write the **rx_buffer_offset** register value.
- Setup the receive channel(s) buffer descriptors in host memory as required by CPPI 3.0.
- Enable the RX DMA controller by setting the **rx_en** bit in the **Rx_Control** register.

13.3.1.1.1.2 Receive Channel Teardown

The host commands a receive channel teardown by writing the channel number to the **Rx_Teardown** register. When a teardown command is issued to an enabled receive channel the following will occur:

- Any current frame in reception will complete normally.
- The **teardown_complete** bit will be set in the next buffer descriptor in the chain

if there is one.

- The channel head descriptor pointer will be cleared to zero

Copyright © 2011, Texas Instruments Incorporated

- A receive interrupt for the channel will be issued to the host.
- The host should acknowledge a teardown interrupt with a 0xfffffffc

acknowledge value

Channel teardown may be commanded on any channel at any time. The host is informed of the teardown completion by the set teardown complete buffer descriptor bit. The port does not clear any channel enables due to a teardown command. A teardown command to an inactive channel issues an interrupt that software should acknowledge with a 0xfffffffc acknowledge value (note that there is no buffer descriptor in this case). Software may read the interrupt acknowledge location to determine if the interrupt was due to a commanded teardown. The read value will be 0xfffffffc if the interrupt was due to a teardown command.

13.3.1.1.2 Transmit DMA Interface.

The Tranmsit DMA is an eight channel CPPI 3.0 compliant interface. Priority between the eight queues may be either fixed or round robin as selected by **tx_ptype** in the **DMAControl** register. If the priority type is fixed, then channel 7 has the highest priority and channel 0 has the lowest priority. Round robin priority proceeds from channel 0 to channel 7.

13.3.1.1.2.1 Transmit DMA Host Configuration

To configure the TX DMA for operation the host must do the following:

- Initialize the **Tx_HDP** registers to a zero value.
- Enable the desired transmit interrupts in the **IntMask** register.
- Setup the transmit channel(s) buffer descriptors in host memory as defined in CPPI 3.0.
- Configure and enable the transmit operation as desired in the **TxControl** register.
- Write the appropriate **Tx_HDP** registers with the appropriate values to start transmit operations.

13.3.1.1.2.2 Transmit Channel Teardown

The host commands a transmit channel teardown by writing the channel number to the **Tx_Teardown** register. When a teardown command is issued to an enabled transmit channel the following will occur:

- Any frame currently in transmission will complete normally
- The teardown complete bit will be set in the next sop buffer descriptor (if there is one).
- The channel head descriptor pointer will be set to zero.
- An interrupt will be issued to inform the host of the channel teardown.
- The host should acknowledge a teardown interrupt with a 0xfffffffc acknowledge value

Channel teardown may be commanded on any channel at any time. The host is informed of the teardown completion by the set teardown complete buffer descriptor bit. The port does not clear any channel enables due to a teardown command. A teardown command to an inactive channel issues an interrupt that software should acknowledge with a 0xfffffffc acknowledge value (note that there is no buffer descriptor in this case). Software may read the interrupt acknowledge location to determine if the interrupt was due to a commanded teardown. The read value will be 0xfffffffc if the interrupt was due to a teardown command.

13.3.1.1.3 Transmit Rate Limiting

Transmit operations can be configured to rate limit the transmit data for each transmit priority. Rate limiting is enabled for a channel when the **tx_rlim[7:0]** bit associated with that channel is set in the **DMAControl** register. Rate limited channels must be the highest priority channels. For example, if two rate limited channels are required then **tx_rlim[7:0]** should be set to 11000000 with the msb corresponding to channel 7. When any channels are configured to be rate-limiting, the priority type must be fixed for transmit. Round-robin priority type is not allowed when rate-limiting. Each of the eight transmit priorities has an

Copyright © 2011, Texas Instruments Incorporated

associated register to control the rate at which the priority is allowed to send data (**Tx_Pri(7..0)_Rate)** when the channel is rate-limiting. Each priority has a send count (**pri(7..0)_send_cnt[13:0]**) and an idle count (**pri(7..0)_idle_cnt[13:0]**). The transfer rate includes the inter-packet gap (12 bytes) and the preamble (8 bytes). The rate in Mbits/second that each priority is allowed to send is controlled by the below equation.

Priority Transfer rate in Mbit/s = ((**priN_idle_cnt**/(**priN_idle_cnt** + **priN_send_cnt**)) * frequency * 32

Where the frequency is the **CLK** frequency.

13.3.1.1.4 Command IDLE

The **cmd_idle** bit in the **DMA_Control** register allows CPDMA operation to be suspended. When the idle state is commanded, the CPDMA will stop processing receive and transmit frames at the next frame boundary. Any frame currently in reception or transmission will be completed normally without suspension. For transmission, any complete or partial frame in the tx cell fifo will be transmitted. For receive, frames that are detected by the CPDMA after the suspend state is entered are ignored. No statistics will be kept for ignored frames. Commanded idle is similar in operation to emulation control and clock stop.

13.3.1.2 Address Lookup Engine (ALE)

The address lookup engine (ALE) processes all received packets to determine which port(s) if any that the packet should the forwarded to. The ALE uses the incoming packet received port number, destination address, source address, length/type, and VLAN information to determine how the packet should be forwarded. The ALE outputs the port mask to the switch fabric that indicates the port(s) the packet should be forwarded to. The ALE is enabled when the ale_enable bit in the ALE_Control register is set. All packets are dropped when the ale_enable bit is cleared to zero.

In normal operation, the CPGMAC_SL modules are configured to issue an abort, instead of an end of packet, at the end of a packet that contains an error (runt, frag, oversize, jabber, crc, alignment, code etc.) or at the end of a mac control packet. However, when the CPGMAC_SL configuration bit(s) cef, csf, or cmf are set, error frames, short frames or mac control frames have a normal end of packet instead of an abort at the end of the packet. When the ALE receives a packet that contains errors (due to a set header error bit), or a mac control frame and does not receive an abort, the packet will be forwarded only to the host port (port 0). No ALE learning occurs on packets with errors or mac control frames. Learning is based on source address and lookup is based on destination address.

The ALE may be configured to operate in bypass mode by setting the ale_bypass bit in the ALE_Control register. When in bypass mode, all CPGMAC_SL received packets are forwarded only to the host port (port 0). Packets from the two ports can be on separate Rx DMA channels by configuring the CPDMA_Rx_Ch_Map register. In bypass mode, the ALE processes host port transmit packets the same as in normal mode. In general, packets would be directed by the host in bypass mode.

The ALE may be configured to operate in OUI deny mode by setting the enable_oui_deny bit in the ALE_Control register. When in OUI deny mode, a packet with a non-matching OUI source address will be dropped unless the destination address matches a multicast table entry with the super bit set. Broadcast packets will be dropped unless the broadcast address is entered into the table with the super bit set. Unicast packets will be dropped unless the unicast address is in the table with block and secure both set (supervisory unicast packet).

Multicast supervisory packets are designated by the super bit in the table entry. Unicast supervisory packets are indicated when block and secure are both set. Supervisory packets are not dropped due to rate limiting, OUI, or VLAN processing.

13.3.1.2.1 Address Table Entry

The ALE table contains 1024 entries. Each table entry represents a free entry, an address, a VLAN, an address/VLAN pair, or an OUI address. Software should ensure that there are not double address entries in the table. The double entry used would be indeterminate. Reserved table bits must be written with zeroes.

Copyright © 2011, Texas Instruments Incorporated

Source Address learning occurs for packets with a unicast, multicast or broadcast destination address and a unicast or multicast (including broadcast) source address. Multicast source addresses have the group bit (bit 40) cleared before ALE processing begins, changing the multicast source address to a unicast source address. A multicast address of all ones is the broadcast address which may be added to the table. A learned unicast source address is added to the table with the following control bits:

Table 13-5. Learned Address Control Bits

unicast_type	11
Block	0
Secure	0

If a received packet has a source address that is equal to the destination address then the following occurs:

- The address is learned if the address is not found in the table.
- The address is updated if the address is found.
- The packet is dropped.

Table Entry Type

00 – Free Entry

01 – Address Entry : unicast or multicast determined by dest **address bit 40 .**

10 – VLAN entry

11 – VLAN Address Entry : unicast or multicast determined by **address bit 40.**

13.3.1.2.1.1 Free Table Entry

Table 13-6. Free (Unused) Address Table Entry Bit Values

71:62	**61:60**	**59:0**
Reserved	Entry Type	Reserved

13.3.1.2.1.2 Multicast Address Table Entry

Table 13-7. Multicast Address Table Entry Bit Values

71:69	**68:66**	**65**	**64**	**63:62**	**61:60**	**59:48**	**47:0**
Reserved	Port Mask	Super	Reserved	Mcast Fwd State	Entry Type	Reserved	Multicast Address

Supervisory Packet (SUPER)

When set, this field indicates that the packet with a matching multicast destination address is a supervisory packet.

0: Non-supervisory packet

1: Supervisory packet

Port Mask(2:0) (PORT_MASK)

This three bit field is the port bit mask that is returned with a found multicast destination address. There may be multiple bits set indicating that the multicast packet may be forwarded to multiple ports (but not the receiving port).

Multicast Forward State (MCAST_FWD_STATE)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Multicast Forward State – Indicates the port state(s) required for the received port on a destination address lookup in order for the multicast packet to be forwarded to the transmit port(s). A transmit port must be in the Forwarding state in order to forward the packet. If the transmit port_mask has multiple set bits then each forward decision is independent of the other transmit port(s) forward decision.

00 – Forwarding

01 – Blocking/Forwarding/Learning

10 – Forwarding/Learning

11 – Forwarding

The forward state test returns a true value if both the Rx and Tx ports are in the required state.

Table Entry Type (ENTRY_TYPE)

Address entry type. Unicast or multicast determined by address bit 40.

01: Address entry. Unicast or multicast determined by address bit 40.

Packet Address (MULTICAST_ADDRESS)

This is the 48-bit packet MAC address. For an OUI address, only the upper 24-bits of the address are used in the source or destination address lookup. Otherwise, all 48-bits are used in the lookup.

13.3.1.2.1.3 VLAN/Multicast Address Table Entry

Table 13-8. VLAN/Multicast Address Table Entry Bit Values

71:69	68:66	65	64	63:62	61:60	59:48	47:0
Reserved	Port Mask	Super	Reserved	Mcast Fwd State	Entry Type (11)	vlan_id	Multicast Address

Supervisory Packet (SUPER)

When set, this field indicates that the packet with a matching multicast destination address is a supervisory packet.

0: Non-supervisory packet

1: Supervisory packet

Port Mask(2:0) (PORT_MASK)

This three bit field is the port bit mask that is returned with a found multicast destination address. There may be multiple bits set indicating that the multicast packet may be forwarded to multiple ports (but not the receiving port).

Multicast Forward State (MCAST_FWD_STATE)

Multicast Forward State – Indicates the port state(s) required for the received port on a destination address lookup in order for the multicast packet to be forwarded to the transmit port(s). A transmit port must be in the Forwarding state in order to forward the packet. If the transmit port_mask has multiple set bits then each forward decision is independent of the other transmit port(s) forward decision.

00 – Forwarding

01 – Blocking/Forwarding/Learning

10 – Forwarding/Learning

11 – Forwarding

The forward state test returns a true value if both the Rx and Tx ports are in the required state.

Table Entry Type (ENTRY_TYPE)

Address entry type. Unicast or multicast determined by address bit 40.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

11: VLAN address entry. Unicast or multicast determined by address bit 40.

VLAN ID (VLAN_ID)

The unique identifier for VLAN identification. This is the 12-bit VLAN ID.

Packet Address (MULTICAST_ADDRESS)

This is the 48-bit packet MAC address. For an OUI address, only the upper 24-bits of the address are used in the source or destination address lookup. Otherwise, all 48-bits are used in the lookup.

13.3.1.2.1.4 Unicast Address Table Entry

Table 13-9. Unicast Address Table Entry Bit Values

71:68	67:66	65	64	63:62	61:60	59:48	47:0
Reserved	Port Number	Block	Secure	Unicast Type (00) or (X1)	Entry Type	Reserved	Unicast Address

Port Number (PORT_NUMBER)

Port Number – This field indicates the port number (not port mask) that the packet with a unicast destination address may be forwarded to. Packets with unicast destination addresses are forwarded only to a single port (but not the receiving port).

Block (BLOCK)

Block – The block bit indicates that a packet with a matching source or destination address should be dropped (block the address).

0 – Address is not blocked.

1 – Drop a packet with a matching source or destination address (secure

must be zero)

If block and secure are both set, then they no longer mean block and secure. When both are set, the block and secure bits indicate that the packet is a unicast supervisory (super) packet and they determine the unicast forward state test criteria. If both bits are set then the packet is forwarded if the receive port is in the Forwarding/Blocking/Learning state. If both bits are not set then the packet is forwarded if the receive port is in the Forwarding state.

Secure (SECURE)

Secure – This bit indicates that a packet with a matching source address should be dropped if the received port number is not equal to the table entry port_number.

0 – Received port number is a don’t care.

1 – Drop the packet if the received port is not the secure port for the source

address and do not update the address (block must be zero)

Unicast Type (UNICAST_TYPE)

Unicast Type – This field indicates the type of unicast address the table entry contains.

00 – Unicast address that is not ageable.

01 – Ageable unicast address that has not been touched.

10 – OUI address - lower 24-bits are don’t cares (not ageable).

11 – Ageable unicast address that has been touched.

Table Entry Type (ENTRY_TYPE)

Address entry. Unicast or multicast determined by address bit 40.

01: Address entry. Unicast or multicast determined by address bit 40.

Copyright © 2011, Texas Instruments Incorporated

Packet Address (UNICAST_ADDRESS)

This is the 48-bit packet MAC address. All 48-bits are used in the lookup.

13.3.1.2.1.5 OUI Unicast Address Table Entry

Table 13-10. OUI Unicast Address Table Entry Bit Values

71:64	63:62	61:60	59:48	47:24	23:0
Reserved	Unicast Type (10)	Entry Type (01)	Reserved	Unicast OUI	Reserved

Unicast Type (UNICAST_TYPE)

Unicast Type – This field indicates the type of unicast address the table entry contains.

00 – Unicast address that is not ageable.

01 – Ageable unicast address that has not been touched.

10 – OUI address - lower 24-bits are don't cares (not ageable).

11 – Ageable unicast address that has been touched.

Table Entry Type (ENTRY_TYPE)

Address entry. Unicast or multicast determined by address bit 40.

01: Address entry. Unicast or multicast determined by address bit 40.

Packet Address (UNICAST_OUI)

For an OUI address, only the upper 24-bits of the address are used in the source or destination address lookup.

13.3.1.2.1.6 VLAN/Unicast Address Table Entry

Table 13-11. Unicast Address Table Entry Bit Values

71:68	67:66	65	64	63:62	61:60	59:48	47:0
Reserved	Port Number	Block	Secure	Unicast Type (00) or (X1)	Entry Type (11)	vlan_id	Unicast Address

Port Number (PORT_NUMBER)

Port Number – This field indicates the port number (not port mask) that the packet with a unicast destination address may be forwarded to. Packets with unicast destination addresses are forwarded only to a single port (but not the receiving port).]

Block (BLOCK)

Block – The block bit indicates that a packet with a matching source or destination address should be dropped (block the address).

0 – Address is not blocked.

1 – Drop a packet with a matching source or destination address (secure

must be zero)

If block and secure are both set, then they no longer mean block and secure. When both are set, the block and secure bits indicate that the packet is a unicast supervisory (super) packet and they determine the unicast forward state test criteria. If both bits are set then the packet is forwarded if the receive port is in the Forwarding/Blocking/Learning state. If both bits are not set then the packet is forwarded if the receive port is in the Forwarding state.

Secure (SECURE)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Secure – This bit indicates that a packet with a matching source address should be dropped if the received port number is not equal to the table entry port_number.

0 – Received port number is a don't care.

1 – Drop the packet if the received port is not the secure port for the source

address and do not update the address (block must be zero)

Unicast Type (UNICAST_TYPE)

Unicast Type – This field indicates the type of unicast address the table entry contains.

00 – Unicast address that is not ageable.

01 – Ageable unicast address that has not been touched.

10 – OUI address - lower 24-bits are don't cares (not ageable).

11 – Ageable unicast address that has been touched.

Table Entry Type (ENTRY_TYPE)

Address entry. Unicast or multicast determined by address bit 40.

11: VLAN address entry. Unicast or multicast determined by address bit 40.

VLAN ID (VLAN_ID)

The unique identifier for VLAN identification. This is the 12-bit VLAN ID.

Packet Address (UNICAST_ADDRESS)

This is the 48-bit packet MAC address. All 48-bits are used in the lookup.

13.3.1.2.1.7 VLAN Table Entry

Table 13-12. VLAN Table Entry

71:62	61:60	59:48	47:27	26:24	23:19	18:16	15:11	10:8	7:3	2:0
Reserved	Entry Type (10)	vlan_id	Reserved	Force Untagged Egress	Reserved	Reg Mcast Flood Mask	Reserved	Unreg Mcast Flood Mask	Reserved	Vlan Member List

Table Entry Type (ENTRY_TYPE)

10: VLAN entry

VLAN ID (VLAN_ID)

The unique identifier for VLAN identification. This is the 12-bit VLAN ID.

Force Untagged Packet Egress (FORCE_UNTAGGED_EGRESS)

This field causes the packet VLAN tag to be removed on egress (except on port 0).

Registered Multicast Flood Mask (REG_MCAST_FLOOD_MASK)

Registered Multicast Flood Mask – Mask used for multicast when the multicast address is found

Unregistered Multicast Flood Mask (UNREG_MCAST_FLOOD_MASK)

Unregistered Multicast Flood Mask – Mask used for multicast when the multicast address is not found

VLAN Member List (VLAN_MEMBER_LIST)

VLAN Member List – This three bit field indicates which port(s) are members of the associated VLAN.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.3.1.2.2 Packet Forwarding Processes

There are four processes that an incoming received packet may go through to determine packet forwarding. The processes are *Ingress Filtering, VLAN_Aware Lookup, VLAN_Unaware Lookup,* and *Egress*.

Packet processing begins in the *Ingress Filtering* process. Each port has an associated packet forwarding state that can be one of four values (Disabled, Blocked, Learning, or Forwarding). The default state for all ports is Disabled. The host sets the packet forwarding state for each port. The receive packet processes

In the packet ingress process (receive packet process), there is a forward state test for unicast destination addresses and a forward state test for multicast addresses. The multicast forward state test indicates the port states required for the receiving port in order for the multicast packet to be forwarded to the transmit port(s). A transmit port must be in the Forwarding state for the packet to be forwarded for transmission. The **mcast_fwd_state** indicates the required port state for the receiving port as indicated in the preceding table. The unicast forward state test indicates the port state required for the receiving port in order to forward the unicast packet. The transmit port must be in the Forwarding state in order to forward the packet. The **block** and **secure** bits determine the unicast forward state test criteria. If both bits are set then the packet is forwarded if the receive port is in the Forwarding/Blocking/Learning state. If both bits are not set then the packet is forwarded if the receive port is in the Forwarding state. The transmit port must be in the Forwarding state regardless. The forward state test used in the ingress process is determined by the destination address packet type (multicast/unicast).

In general, packets received with errors are dropped by the address lookup engine without learning, updating, or touching the address. The error condition and the abort are indicated by the CPGMAC_SL to the ALE. Packets with errors may be passed to the host (not aborted) by a CPGMAC_SL port if the port has a set **rx_cmf_en**, **rx_cef_en**, or **rx_csf_en** bit(s). Error packets that are passed to the host by the CPGMAC_SL are considered to be bypass packets by the ALE and are sent only to the host. Error packets do not learn, update, or touch addresses regardless of whether they are aborted or sent to the host. Packets with errors received by the host are forwarded as normal.

The below control bits are in the SL1/2_**MacControl** register.

rx_cef_en – This CPGMAC_SL control bit enables frames that are fragments, long, jabber, CRC, code, and alignment errors to be forwarded.

rx_csf_en – This CPGMAC_SL bit enables short frames to be forwarded.

rx_cmf_en – This CPGMAC_SL control bit enables mac control frames to be forwarded.

Please refer **TBD Document** on the configuration of Ingress *Filtering, VLAN_Aware Lookup, VLAN_Unaware Lookup,* and *Egress Process*.

13.3.1.2.3 Learning Process

The learning process is applied to each receive packet that is not aborted. The learning process is a concurrent process with the packet forwarding process.

Please refer to **TBD Document** for details on the configuration of Learnig Process

13.3.1.2.4 VLAN Aware Mode

The CPSW_3G is in VLAN aware mode when the CPSW Control register **vlan_aware** bit is set. In VLAN aware mode ports 0 receive packets (out of the CPSW_3G) may or may not be VLAN encapsulated depending on the CPSW Control register rx_vlan_encap bit. Port 0 receive packet data is never modified. VLAN is not removed regardless of the force untagged egress bit for Port 0. VLAN encapsulated receive packets have a 32-bit VLAN header encapsulation word added to the packet data.VLAN encapsulated packets are specified by a set **rx_vlan_encap** bit in the packet buffer descriptor.

Port 0 transmit packets are never VLAN encapsulated (encapsulation is not allowed).

In VLAN aware mode, transmitted packet data is changed depending on the packet type (**pkt_type**), packet priority (**pkt_pri**), and VLAN information as shown in the below tables:

Copyright © 2011, Texas Instruments Incorporated

13.3.1.2.5 VLAN Unaware Mode

The CPSW_3G is in VLAN unaware mode when the CPSW Control register **vlan_aware** bit is cleared. Port 0 receive packets (out of the CPSW_3G) may or may not be VLAN encapsulated depending on the CPSW Control register **rx_vlan_encap** bit. Port 0 transmit packets are never VLAN encapsulated.

13.3.1.3 Packet Priority Handling

Packets are received on three ports, two of which are CPGMAC_SL Ethernet ports and the third port is the CPPI host port. Received packets have a received packet priority (0 to 7 with 7 being the highest priority).

The received packet priority is the port priority for untagged packets, and the actual packet priority for priority tagged and VLAN tagged packets. The received packet priority is mapped through the receive ports associated “packet priority to header packet priority mapping register” to obtain the header packet priority (the CPDMA Rx and Tx nomenclature is reversed from the CPGMAC_SL nomenclature).

The header packet priority is mapped through the “header priority to switch priority mapping register” to obtain the hardware switch priority (0 to 3 with 3 being the highest priority). The header packet priority is then used as the actual transmit packet priority if the VLAN information is to be sent on egress.

13.3.1.4 FIFO Memory Control

Each of the three CPSW_3G ports has an identical associated FIFO. Each FIFO contains a single logical receive queue and four logical transmit queues (priority 0 through 3). Each FIFO memory contains 20,480 bytes (20k) total organized as 2560 by 64-bit words contained in a single memory instance. The FIFO memory is used for the associated port transmit and receive queues. The tx_max_blks field in the FIFO’s associated Max_Blks register determines the maximum number of 1k FIFO memory blocks to be allocated to the four logical transmit queues (transmit total). The rx_max_blks field in the FIFO’s associated Max_Blks register determines the maximum number of 1k memory blocks to be allocated to the logical receive queue. The tx_max_blks value plus the rx_max_blks value must sum to 20 (the total number of blocks in the FIFO). If the sum were less than 20 then some memory blocks would be unused.The default is 17 (decimal) transmit blocks and three receive blocks. The FIFO’s follow the naming convention of the Ethernet ports.Host Port is Port0 and External Ports are Port1,2

13.3.1.5 FIFO Transmit Queue Control

There are four transmit queues in each transmit FIFO. Software has some flexibility in determining how packets are loaded into the queues and on how packet priorities are selected for transmission (how packets are removed and transmitted from queues). All ports on the switch have identical FIFO’s. For the purposes of the below the transmit FIFO is switch egress even though the port 0 transmit FIFO is connected to the CPDMA receive (also switch egress). The CPDMA nomenclature is reversed from the CPGMAC_SL nomenclature due to legacy reasons.

13.3.1.5.1 Normal Priority Mode

When operating in normal mode, lower priority frames are dropped before higher priority frames. The intention is to give preference to higher priority frames. Priority 3 is the highest priority and is allowed to fill the FIFO. Priority 2 will drop packets if the packet is going to take space in the last 2k available. Priority 1 will drop packets if the packet is going to take space in the last 4k available. Priority 0 will drop packets if the packet is going to take space in the last 6k available. If fewer than 4 priorities are to be implemented then the priorities should be mapped such that the highest priorities are used. For example, if two priorities are going to be used then all packets should be mapped to priorities 3 and 2 and priorities 1 and 0 should be unused. Priority escalation may be used in normal priority mode if desired. Normal priority mode is configured as described below:

- Select normal priority mode by setting tx_in_sel[1:0] = 00 for all ports (default value in P0/1/2_Tx_In_Ctl)
- Configure priority mapping to use only the highest priorities if less than 4 priorities are used. Refer to the Packet Priority Handling section of this specification.

Copyright © 2011, Texas Instruments Incorporated

13.3.1.5.2 Dual Mac Mode

When operating in dual mac mode the intention is to transfer packets between ports 0 and 1 and ports 0 and 2, but not between ports 1 and 2. Each CPGMAC_SL appears as a single MAC with no bridging between MAC's. Each CPGMAC_SL has at least one unique (not the same) mac address.

Dual mac mode is configured as described below:

- Set the ale_vlan_aware bit in the ALE_Control register. This bit configures the ALE to process in vlan aware mode.The CPSW_3G vlan aware bit (vlan_aware in CPSW_Control) determines how packets VLAN's are processed on CPGMAC_SL egress and does not affect how the ALE processes packets or the packet destination. The CPSW_3G vlan aware bit may be set or not as required (must be set if VLAN's are to exit the switch).
- **Configure the Port 1 to Port 0 VLAN**

Add a VLAN Table Entry with ports 0 and 1 as members (clear the flood masks).

Add a VLAN/Unicast Address Table Entry with the Port1/0 VLAN and a port number of 0. Packets received on port 1 with this unicast address will be sent only to port 0 (egress). If multiple mac addresses are desired for this port then multiple entries of this type may be configured.

- **Configure the Port 2 to Port 0 VLAN**

Add a VLAN Table Entry with ports 0 and 2 as members (clear the flood masks).

Add a VLAN/Unicast Address Table Entry with the Port2/0 VLAN and a port number of 0. Packets received on port 2 with this unicast address will be sent only to port 0 (egress). If multiple mac addresses are desired for this port then multiple entries of this type may be configured.

- Packets from the host (port 0) to ports 1 and 2 should be directed. If directed packets are not desired then VLAN with addresses can be added for both destination ports.
- Select the dual mac mode on the port 0 FIFO by setting tx_in_sel[1:0] = 01 in P0_Tx_In_Ctl. The intention of this mode is to allow packets from both ethernet ports to be written into the FIFO without one port starving the other port.
- The priority levels may be configured such that packets received on port 1 egress on one CPDMA RX channel while packets received on port 2 egress on a different CPDMA RX channel.

13.3.1.6 Audio Video Bridging

AV Bridging is an ongoing project of IEEE 802.1 concerned with enabling low-latency streaming of time-sensitive audiovisual data over networks. Devices are designated as talkers (transmitters), bridges, or listeners (receivers). It is suggested that the maximum latency could be 2 ms over

7 hops for Class A devices and 20 ms over 7 hops for Class B devices. A hop is essentially a single local area network stage in the journey of a packet. Every time a bridge is encountered between one network section and another a hop is involved. One of the performance goals is that AVB streams will not use more than 75 percent of a link's bandwidth, leaving the remaining capacity for non-AVB streams.

The goal of developing AVB is simply--extend Ethernet's data-networking capabilities to the realm of reliable real-time audio/video networking.

An "Audio Video Bridging" network is one that implements a set of protocols being developed by the IEEE 802.1 Audio/Video Bridging Task Group. There are four primary differences between the proposed Audio Video Bridging architecture and existing 802 architectures (from now on the term "AVB" will be used instead of "Audio Video Bridging"):

- Precise synchronization - **IEEE 802.1AS:** " *Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks.* " a.k. a **Precision Time Protocol(PTP).**
- Traffic shaping for media streams - IEEE 802.1Qav: "*Virtual Bridged Local Area Networks:* Forwarding and Queuing for Time-Sensitive Streams."
- Admission controls - **IEEE 802.1Qat:** " *Virtual Bridged Local Area Networks - Amendment 9:* **Stream Reservation Protocol (SRP)**".
- Identification of non-participating devices - **IEEE 802.1BA:** "**Audio/Video Bridging (AVB) Systems**"

Figure 13-3. The Network static with AVB

Copyright © 2011, Texas Instruments Incorporated

The media transport protocols that work within the AVB framework are IEEE 1722: Layer 2 Transport Protocol for Time-Sensitive Streams and IEEE 1733: extends RTCP for RTP streaming over AVB-supported networks.

13.3.1.6.1 IEEE 1722: Layer 2 Transport Protocol for Time-Sensitive Streams

AVBTP or 1722 sits above the IEEE 802.1 AVB plumbing and below the application layer. It acts as the conduit between an Ethernet MAC and a streaming application.

AVBTP abstracts the underlying network transmission channel to enable the virtual connection of distributed audio and video CODECs over reliable Ethernet networks.

A complete AVBTP Ethernet packet is shown below. The below figure illustrates how IEC 61883-6 AM824 uncompressed audio samples are encapsulated in an Ethernet frame.

Figure 13-4. IEEE 1722 Packets

13.3.1.6.2 IEEE 1733: extends RTCP for RTP streaming over AVB-supported networks

This standard specifies the protocol, data encapsulations, connection management and presentation time procedures used to ensure interoperability between audio and video based end stations that use standard networking services provided by all IEEE 802 networks meeting QoS requirements for time-sensitive applications by leveraging the Real-time Transport Protocol (RTP) family of protocols and IEEE 802.1 Audio/Video Bridging (AVB) protocols.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.3.1.7 Ethernet Mac Sliver (CPGMAC_SL)

The CPGMAC_SL peripheral shall be compliant to the IEEE Std 802.3 Specification. Half duplex mode is supported in 10/100 Mbps mode, but not in 1000 Mbps (gigabit) mode.

Features:

- Synchronous 10/100/1000 Mbit operation.
- G/MII Interface.
- Hardware Error handling including CRC.
- Full Duplex Gigabit operation (half duplex gigabit is not supported).
- EtherStats and 802.3Stats RMON statistics gathering support for external statistics collection module.
- Transmit CRC generation selectable on a per channel basis.
- Emulation Support.
- VLAN Aware Mode Support.
- Hardware flow control.
- Programmable Inter Packet Gap (IPG)

13.3.1.7.1 GMII/MII Media Independent Interface

The following sections cover operation of the Media Independent Interface in 10/100/1000 Mbps modes. An IEEE 802.3 compliant Ethernet MAC controls the interface.

13.3.1.7.1.1 Data Reception

13.3.1.7.1.1.1 Receive Control

Data received from the PHY is interpreted and output. Interpretation involves detection and removal of the preamble and start of frame delimiter, extraction of the address and frame length, data handling, error checking and reporting, cyclic redundancy checking (CRC), and statistics control signal generation.

13.3.1.7.1.1.2 Receive Inter-Frame Interval

The 802.3 required inter-packet gap (IPG) is 24 GMII clocks (96 bit times) for 10/100 Mbit modes, and 12 GMII clocks (96 bit times) for 1000 Mbit mode. However, the MAC can tolerate a reduced IPG (2 GMII clocks in 10/100 mode and 5 GMII clocks in 1000 mode) with a correct preamble and start frame delimiter.

This interval between frames must comprise (in the following order):

- An Inter-Packet Gap (IPG).
- A seven octet preamble (all octets 0x55).
- A one octet start frame delimiter (0x5D).

13.3.1.7.1.2 Data Transmission

The Gigabit Ethernet Mac Sliver (GMII) passes data to the PHY when enabled. Data is synchronized to the transmit clock rate. The smallest frame that can be sent is two bytes of data with four bytes of CRC (6 byte frame).

13.3.1.7.1.2.1 Transmit Control

A jam sequence is output if a collision is detected on a transmit packet. If the collision was late (after the first 64 bytes have been transmitted) the collision is ignored. If the collision is not late, the controller will back off before retrying the frame transmission. When operating in full duplex mode the carrier sense (CRS) and collision sensing modes are disabled.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.3.1.7.1.2.2 CRC Insertion

The MAC generates and appends a 32-bit Ethernet CRC onto the transmitted data if the transmit packet header **pass_crc** bit is zero. For the CPMAC_SL generated CRC case, a CRC at the end of the input packet data is not allowed.

If the header word **pass_crc** bit is set, then the last four bytes of the TX data are transmitted as the frame CRC. The four CRC data bytes should be the last four bytes of the frame and should be included in the packet byte count value. The MAC performs no error checking on the outgoing CRC when the **pass_crc** bit is set.

13.3.1.7.1.2.3 MTXER

The GMII_MTXER signal is not used. If an underflow condition occurs on a transmitted frame, the frame CRC will be inverted to indicate the error to the network. Underflow is a hardware error.

13.3.1.7.1.2.4 Adaptive Performance Optimization (APO)

The Ethernet MAC port incorporates Adaptive Performance Optimization (APO) logic that may be enabled by setting the **tx_pace** bit in the **MacControl** register. Transmission pacing to enhance performance is enabled when set. Adaptive performance pacing introduces delays into the normal transmission of frames, delaying transmission attempts between stations, reducing the probability of collisions occurring during heavy traffic (as indicated by frame deferrals and collisions) thereby increasing the chance of successful transmission.

When a frame is deferred, suffers a single collision, multiple collisions or excessive collisions, the pacing counter is loaded with an initial value of 31. When a frame is transmitted successfully (without experiencing a deferral, single collision, multiple collision or excessive collision) the pacing counter is decremented by one, down to zero.

With pacing enabled, a new frame is permitted to immediately (after one IPG) attempt transmission only if the pacing counter is zero. If the pacing counter is non zero, the frame is delayed by the pacing delay, a delay of approximately four inter-packet gap delays. APO only affects the IPG preceding the first attempt at transmitting a frame. It does not affect the back-off algorithm for retransmitted frames.

13.3.1.7.1.2.5 Inter-Packet-Gap Enforcement

The measurement reference for the IPG of 96 bit times is changed depending on frame traffic conditions. If a frame is successfully transmitted without collision, and **MCRS** is de-asserted within approximately 48 bit times of **MTXEN** being de-asserted, then 96 bit times is measured from **MTXEN**. If the frame suffered a collision, or if **MCRS** is not de-asserted until more than approximately 48 bit times after **MTXEN** is de-asserted, then 96 bit times (approximately, but not less) is measured from **MCRS**.

The transmit IPG can be shortened by eight bit times when enabled and triggered. The **tx_short_gap_en** bit in the **MacControl** register enables the **TX_SHORT_GAP** input to determine whether the transmit IPG is shorted by eight bit times.

13.3.1.7.1.2.6 Back Off

The Gigabit Ethernet Mac Sliver (GMII) implements the 802.3 binary exponential back-off algorithm.

13.3.1.7.1.2.7 Programmable Transmit Inter-Packet Gap

The transmit inter-packet gap (IPG) is programmable through the **Tx_Gap** register. The default value is decimal 12. The transmit IPG may be increased to the maximum value of 0x1ff. Increasing the IPG is not compatible with transmit pacing. The short gap feature will override the increased gap value, so the short gap feature may not be compatible with an increased IPG.

13.3.1.7.1.2.8 Speed, Duplex and Pause Frame Support Negotiation

The CPMAC_SL can operate in half duplex or full duplex in 10/100 Mbit modes, and can operate in full duplex only in 1000 Mbit mode. Pause frame support is included in 10/100/1000 Mbit modes as configured by the host.

Copyright © 2011, Texas Instruments Incorporated

13.3.1.7.1.3 GMII Interface Signal Connections and Descriptions

GMII Interface can operate in GIG/MII Modes.

In GIG(1000Mbps) Mode 3PSW operates only in Full duplex Mode.

In MII Mode(100/10 Mbps) 3PSW operates in Full duplex and Half Duplex.

The pin connections of the GMII Interace is shown in the following figure.

Figure 13-5. GMII Interface Connections

The detailed description of the signals in GIG/MII Mode are explained in the following sections.

13.3.1.7.1.3.1 GMII Interface Signal Descriptions in GIG (1000Mbps) Mode

Table 13-13. GMII Interface Signal Descriptions in GIG (1000Mbps) Mode

Signal	Type	Description
GMTCLK	O	The transmit clock is a continuous clock that provides the timing reference for transmit operations. The clock is generated by the CPSW and is 125 MHz at 1000 Mbps operation.
MtCLK	I	The transmit clock is a continuous free running clock and is generated by the PHY.
Mtxd	O	The transmit data pins are a collection of 8 data signals comprising 8 bits of data. MTXD0 is the least-significant bit (LSB). The signals are synchronized by GMTCLK and valid only when MTXEN is asserted.
Mtxen	O	The transmit enable signal indicates that the MTXD pins are generating 8bit data for use by the PHY. It is driven synchronously by GMTCLK.
MCOL	I	In full-duplex operation, the MCOL pin is used for hardware transmit flow control. Asserting the MCOL pin will stop packet transmissions; packets in the process of being transmitted when MCOL is asserted will complete transmission. The MCOL pin should be held low if hardware transmit flow control is not used.
MCRS	I	In full-duplex operation, the MCRS pin should be held low.
MRCLK	I	The receive clock is a continuous clock that provides the timing reference for receive operations. The MRXD,MRXDV, and MRXER signals are tied to this clock.The clock is generated by the PHY and is 125 MHz at 1000Mbps of operation.
MRXD	I	The receive data pins are a collection of 8 data signals comprising 8 bits of data.MRXD0 is the least-significant bit (LSB).The signals are synchronized by MRCLK and valid only when MRXDV is asserted.
MRXDV	I	The receive data valid signal indicates that the MRXD pins are generating byte data for use by the 3PSW. It is driven synchronously to MRCLK.
MDCLK	O	Management data clock (MDIO_CLK). The MDIO data clock is sourced by the MDIO module on the system. It is used to synchronize MDIO data access operations done on the MDIO pin.
MDIO	I/O	MDIO DATA(MDI0_D). MDIO data pin drives PHY management data into and out of the PHY by way of an access frame consisting of start of frame, read/write indication,PHY address, register address, and data bit cycles. The MDIO_D pin acts as an output for all but the data bit cycles at which time it is an input for read operations.

13.3.1.7.1.3.2

Table 13-14. GMII Interface Signal Descriptions in MII (100/10Mbps) Mode

Signal	Type	Description
GMTCLK	O	The clock is generated by the CPSW and is running at 125 MHz
MtCLK	I	The transmit clock is a continuous clock that provides the timing reference for transmit operations. The MTXD and MTXEN signals are tied to this clock. The clock is generated by the PHY and is 2.5 MHz at 10 Mbps operation and 25 MHz at 100 Mbps operation.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 13-14. GMII Interface Signal Descriptions in MII (100/10Mbps) Mode (continued)

Signal	Type	Description
Mtxd	O	MTXD[7-4] pins of MTXD data are not used.The transmit data pins are a collection of 4 data signals MTXD[3-0] comprising 4 bits of data. MTXD0 is the least-significant bit (LSB). The signals are synchronized by MTCLK and valid only when MTXEN is asserted.
Mtxen	O	The transmit enable signal indicates that the MTXD pins are generating 4bit data for use by the PHY. It is driven synchronously by MTCLK.
MCOL	I	In half-duplex operation, the MCOL pin is asserted by the PHY when it detects a collision on the network. It remains asserted while the collision condition persists.This signal is not necessarily synchronous to MTCLK nor MRCLK. In full-duplex operation, the MCOL pin is used for hardware transmit flow control. Asserting the MCOL pin will stop packet transmissions; packets in the process of being transmitted when MCOL is asserted will complete transmission. The MCOL pin should be held low if hardware transmit flow control is not used
MCRS	I	In half-duplex operation, the MCRS pin is asserted by the PHY when the network is not idle in either transmit or receive. The pin is deasserted when both transmit and receive are idle. This signal is not necessarily synchronous to MTCLK nor MRCLK. In full-duplex operation, the MCRS pin should be held low.
MRCLK	I	The receive clock is a continuous clock that provides the timing reference for receive operations. The MRXD, MRXDV, and MRXER signals are tied to this clock. The clock is generated by the PHY and is 2.5 MHz at 10 Mbps operation and 25 MHz at 100 Mbps operation.
MRXD	I	MRXD[7-4] bits are unused in the MII mode**.**The receive data pins are a collection of 4 data signals comprising 4 bits of data.MRXD0 is the least-significant bit (LSB).The signals are synchronized by MRCLK and valid only when MRXDV is asserted.
MRXDV	I	The receive data valid signal indicates that the MRXD pins are generating nibble data for use by the 3PSW. It is driven synchronously to MRCLK.
MDCLK	O	Management data clock (MDIO_CLK). The MDIO data clock is sourced by the MDIO module on the system. It is used to synchronize MDIO data access operations done on the MDIO pin.
MDIO	I/O	MDIO DATA(MDI0_D). MDIO data pin drives PHY management data into and out of the PHY by way of an access frame consisting of start of frame, read/write indication,PHY address, register address, and data bit cycles. The MDIO_D pin acts as an output for all but the data bit cycles at which time it is an input for read operations.

13.3.1.7.2 RMII Interface

The CPRMII peripheral shall be compliant to the RMII specification document.

Features:

- Source Synchronous 10/100 Mbit operation.
- Full and Half Duplex support.

13.3.1.7.2.1 RMII Receive (RX)

The CPRMII receive (RX) interface converts the input data from the external RMII PHY (or switch) into the required MII (CPGMAC) signals. The carrier sense and collision signals are determined from the RMII input data stream and transmit inputs as defined in the RMII specification.

An asserted **RMRXER** on any di-bit in the received packet will cause an **MRXER** assertion to the CPGMAC during the packet. In 10Mbps mode, the error is not required to be duplicated on 10 successive clocks. Any di-bit which has an asserted RMII_RXER during any of the 10 replications of the data will cause the error to be propagated.

Any received packet that ends with an improper nibble boundary aligned **RMCRSDV** toggle will issue an **MRXER** during the packet to the CPGMAC. Also, a change in speed or duplex mode during packet operations will cause packet corruption.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

The CPRMII can accept receive packets with shortened preambles, but 0x55 followed by a 0x5d is the shortest preamble that will be recognized (1 preamble byte with the start of frame byte). At least one byte of preamble with the start of frame indicator is required to begin a packet. An asserted **RMCRSDV** without at least a single correct preamble byte followed by the start of frame indicator will be ignored.

13.3.1.7.2.2 RMII Transmit (TX)

The CPRMII transmit (TX) interface converts the 3PSW MII input data into the RMII transmit format. The data is then output to the external RMII PHY.

The 3PSW does not source the transmit error (MII TXERR) signal. Any transmit frame from the CPGMAC with an error (ie. underrun) will be indicated as an error by an error CRC. Transmit error is assumed to be deasserted at all times and is not an input into the CPRMII module.Zeroes are output on RMTXD[1:0] for each clock that RMTXEN is deasserted.

13.3.1.7.2.3 RMII Signal Connections and Descriptions

The following figure shows a device with integrated 3PSW and MDIO interfaced via a RMII connection in a typical system.

The individual CPSW and MDIO signals for the RMII interface are summarized in the following figure.

For more information, see either the IEEE 802.3 standard or ISO/IEC 8802-3:2000(E).

Figure 13-6. RMII Interface Connections

Table 13-15.

Signal	Type	Description
RMTXD [1-0]	O	Transmit data (RMII_TXD). The transmit data pins are a collection of 2 bits of data. RMTDX0 is the least-significant bit (LSB). The signals are synchronized by RMREFCLK and valid only when RMTXEN is asserted.
RMXEN	O	Transmit enable (RMII_TXEN). The transmit enable signal indicates that the RMII_TXD pins are generating data for use by the PHY. RMII_TXEN is synchronous to RMII_MHZ_50_CLK.
RMREFCLK	I/O	RMII reference clock. (RMII_MHZ_50_CLK) The reference clock is used to synchronize all RMII signals. RMREFCLK must be continuous and fixed at 50 MHz.
RMRXD [1-0]	I	Receive data (RMII_RXD). The receive data pins are a collection of 2 bits of data. RMRDX0 is the least-significant bit (LSB). The signals are synchronized by RMREFCLK and valid only when RMCRSDV is asserted and RMRXER is deasserted.
RMCRSDV	I	Carrier sense/receive data valid (RMCRSDV). Multiplexed signal between carrier sense and receive data valid.
RMRXER	I	Receive error (RMII_RXER). The receive error signal is asserted to indicate that an error was detected in the received frame.
MDCLK	O	Management data clock (MDIO_CLK). The MDIO data clock is sourced by the MDIO module on the system. It is used to synchronize MDIO data access operations done on the MDIO pin.
MDIO	I/O	MDIO DATA(MDI0_D). MDIO data pin drives PHY management data into and out of the PHY by way of an access frame consisting of start of frame, read/write indication,PHY address, register address, and data bit cycles. The MDIO_D pin acts as an output for all but the data bit cycles at which time it is an input for read operations.

13.3.1.7.3 RGMII Interface

The CPRGMII peripheral shall be compliant to the RGMII specification document.

Features:

- Supports 1000/100/10 Mbps Speed.
- MII mode is not supported.

Copyright © 2011, Texas Instruments Incorporated

- Selectable internal delay on transmit.

13.3.1.7.3.1 RGMII Receive (RX)

The CPRGMII receive (RX) interface converts the source synchronous DDR input data from the external RGMII PHY into the required G/MII (CPGMAC) signals.

13.3.1.7.3.2 In-Band Mode of Operation

The CPRGMII is operating in the in-band mode of operation when the **RGMII_RX_INBAND** input is asserted.RGMII_RX_INPUT is asserted by configuring the ext_en bit to 1 of sl_mac_control register.The link status, duplexity, and speed are determined from the RGMII input data stream as defined in the RGMII specification. The link speed is indicated as shown in the following table:

RGMII_SPEED(1:0)	Link Speed
00	10 Mbs mode
01	100 Mbs mode
10	1000 Mbs mode
11	reserved

13.3.1.7.3.3 Forced Mode of Operation

The CPRGMII is operating in the forced mode of operation when the **RGMII_RX_INBAND** input is deasserted. In the forced mode of operation, the in-band data is ignored if present. The link status is forced high, and the duplexity and speed are determined from the **RGMII_FULLDUPLEX_IN** and **RGMII_GIG_IN** inputs. If the **RGMII_GIG_IN** input is asserted, then operation is gigabit mode (**RGMII_SPEED (1:0)** = 10). If the **RGMII_GIG_IN** input is deasserted, the operation is 100mbps mode (**RGMII_SPEED (1:0)** = 01).

13.3.1.7.3.4 RGMII Transmit (TX)

The CPRGMII transmit (TX) interface converts the CPGMAC G/MII input data into the DDR RGMII format. The DDR data is then output to the external PHY.

The CPGMAC does not source the transmit error (MTXERR) signal. Any transmit frame from the CPGMAC with an error (ie. underrun) will be indicated as an error by an error CRC. Transmit error is assumed to be deasserted at all times and is not an input into the CPRGMII module.

In 10/100 mode, the MTXD (7:0) data bus uses only the lower nibble. The CPRGMII will output the lower nibble twice in 10/100 mode to avoid unnecessary signal switching.

Packets will be precluded from transmission through the CPRGMII module for 4096 transmit clocks after the rising edge of **RGMII_LINK**. Packet transmission will begin on the first **GMII_MTXEN** rising edge after the 4096 transmit clock count has expired.

The RGMII0/1_ID_MODE bit value in the GMII_SEL register determines whether or not the transmit delay is included in the CPRGMII or not. When RGMII0/1_ID_MODE bit is zero, the transmit delay is included. The RGMII0/1_ID_MODE input is a configuration input only and is not intended to be changed during packet operations.

Please refer to the **Device Control Module** Document, **MII mode selection register (GMII_SEL)** to configure the CPRGMII Internal Delay/No Delay in Transmit Mode of Operation.

13.3.1.7.3.5 RGMII Signal Connections and Descriptions

The below figure shows a device with integrated CPSW and MDIO interfaced via a RGMII connection in a typical system.

The individual CPSW and MDIO signals for the RGMII interface are summarized in Figur6.

Figure 13-7. RGMII Interface Connections

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 13-16.

Signal	Type	Description
RGTXD [3-0]	O	The transmit data pins are a collection of 4 bits of data.RGTDX0 is the least-significant bit (LSB). The signals are valid only when RGTXCTL is asserted.
RGTXCTL	O	Transmit Control/enable .The transmit enable signal indicates that the RGTXD pins are generating data for use by the PHY.
RGTXC	O	The transmit reference clock will be 125Mhz, 25Mhz, or 2.5Mhz depending on speed of operation.
RGRXD [3-0]	I	The receive data pins are a collection of 4 bits of data. RGRDX0 is the least-significant bit (LSB). The signals are valid only when RGRXCTL is asserted.
RGRXCTL	I	The receive data valid/control signal indicates that the RGRXD pins are nibble data for use by the 3PSW.
RGRXC	I	The receive clock is a continuous clock that provides the timing reference for receive operations.The clock is generated by the PHY and is 2.5 MHz at 10 Mbps operation and 25 MHz at 100 Mbps operation,125 MHz at 1000Mbps of operation.
MDCLK	O	Management data clock (MDIO_CLK). The MDIO data clock is sourced by the MDIO module on the system. It is used to synchronize MDIO data access operations done on the MDIO pin.
MDIO	I/O	MDIO DATA(MDI0_D). MDIO data pin drives PHY management data into and out of the PHY by way of an access frame consisting of start of frame, read/write indication,PHY address, register address, and data bit cycles. The MDIO_D pin acts as an output for all but the data bit cycles at which time it is an input for read operations.

13.3.1.7.4 Frame Classification

Received frames are proper (good) frames if they are between 64 and **rx_maxlen** in length (inclusive) and contain no errors (code/align/CRC).

Received frames are long frames if their frame count exceeds the value in the **rx_maxlen** register. The **rx_maxlen** register reset (default) value is 1518 (dec). Long received frames are either oversized or jabber frames. Long frames with no errors are oversized frames. Long frames with CRC, code, or alignment errors are jabber frames.

Received frames are short frames if their frame count is less than 64 bytes. Short frames that contain no errors are undersized frames. Short frames with CRC, code, or alignment errors are fragment frames.

A received long packet will always contain **rx_maxlen** number of bytes transferred to memory (if **rx_cef_en** = 1). An example with **rx_maxlen** = 1518 is below:

- If the frame length is 1518, then the packet is not a long packet and there will be 1518 bytes transferred to memory.
- If the frame length is 1519, there will be 1518 bytes transferred to memory. The last three bytes will be the first three CRC bytes.
- If the frame length is 1520, there will be 1518 bytes transferred to memory. The last two bytes will be the first two CRC bytes.
- If the frame length is 1521, there will be 1518 bytes transferred to memory. The last byte will be the first CRC byte.

If the frame length is 1522, there will be 1518 bytes transferred to memory. The last byte will be the last data byte.

13.3.1.8 Embedded Memories

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Memory Type Description	Number of Instances
Two port 36-word by 32-bit RAM	1 (STATISTICS)
Single port 2560 by 64 RAM	3 (Packet FIFOs)
Single port 64-word by 1152-bit RAM	1 (ALE)
Single port 2048-word by 32-bit RAM	1 (CPPI)

13.3.1.9 Flow Control

There are two types of switch flow control – CPPI port flow control and Ethernet port flow control. The CPPI and Ethernet port naming conventions for data flow into and out of the switch are reversed. For the CPPI port (port 0), transmit operations move packets from external memory into the switch and then out to either or both Ethernet transmit ports (ports 1 and 2). CPPI receive operations move packets that were received on either or both Ethernet receive ports to external memory.

13.3.1.9.1 CPPI Port Flow Control

The CPPI port has flow control available for transmit (switch ingress). CPPI receive operations (switch egress) do not require flow control. CPPI Transmit flow control is initiated when enabled and triggered. CPPI transmit flow control is enabled by setting the p0_flow_en bit in the **CPSW_Flow_Control** register. CPPI transmit flow control is enabled by default on reset because host packets should not be dropped in any mode of operation.

13.3.1.9.2 Ethernet Port Flow Control

The Ethernet ports have flow control available for transmit and receive. Transmit flow control stops the Ethernet port from transmitting packets to the wire (switch egress) in response to a received pause frame. Transmit flow control does not depend on FIFO usage.

The ethernet ports have flow control available for receive operations (packet ingress). Ethernet port receive flow control is initiated when enabled and triggered. Packets received on an ethernet port can be sent to the other ethernet port or the CPPI port (or both). Each destination port can trigger the receive ethernet port flow control. An ethernet destination port triggers another ethernet receive flow control when the destination port is full.

When a packet is received on an ethernet port interface with enabled flow control the below occurs:

- The packet will be sent to all ports that currently have room to take the entire packet.
- The packet will be retried until successful to all ports that indicate they don't have room for the packet.

The flow control trigger to the CPGMAC_SL will be asserted until the packet has been sent, and there is room in the logical receive FIFO for packet runout from another flow control trigger (**rx_pkt_cnt** = 0). Ethernet port receive flow control is disabled by default on reset. Ethernet port receive flow control requires that the **rx_flow_en** bit in the associated CPGMAC_SL be set to one. When receive flow control is enabled on a port, the port's associated FIFO block allocation must be adjusted. The port RX allocation must increase from the default three blocks to accommodate the flow control runout. A corresponding decrease in the TX block allocation is required. If a sending port ignores a pause frame then packets may overrun on receive (and be dropped) but will not be dropped on transmit. If flow control is disabled for gmii ports, then any packets that are dropped are dropped on transmit and not on receive.

13.3.1.9.2.1 Receive Flow Control

When enabled and triggered, receive flow control is initiated to limit the CPGMAC_SL from further frame reception. Half-duplex mode receive flow control is collision based while full duplex mode issues 802.3X pause frames. In either case, receive flow control prevents frame reception by issuing the flow control appropriate for the current mode of operation. Receive flow control is enabled by the **rx_flow_en** bit in the **MacControl** register. Receive flow control is triggered (when enabled) when the **RX_FLOW_TRIGGER** input is asserted. The CPGMAC_SL is configured for collision or IEEE 802.3X flow control via the **fullduplex** bit in the **MacControl** register.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.3.1.9.2.1.1 Collision Based Receive Buffer Flow Control

Collision-based receive buffer flow control provides a means of preventing frame reception when the port is operating in half-duplex mode (**fullduplex** is cleared in **MacControl)**. When receive flow control is enabled and triggered, the port will generate collisions for received frames. The jam sequence transmitted will be the twelve byte sequence C3.C3.C3.C3.C3.C3.C3.C3.C3.C3.C3.C3 (hex). The jam sequence will begin no later than approximately as the source address starts to be received. Note that these forced collisions will not be limited to a maximum of 16 consecutive collisions, and are independent of the normal back-off algorithm. Receive flow control does not depend on the value of the incoming frame destination address. A collision will be generated for any incoming packet, regardless of the destination address.

13.3.1.9.2.1.2 IEEE 802.3X Based Receive Flow Control

IEEE 802.3x based receive flow control provides a means of preventing frame reception when the port is operating in full-duplex mode (**fullduplex** is set in **MacControl)**. When receive flow control is enabled and triggered, the port will transmit a pause frame to request that the sending station stop transmitting for the period indicated within the transmitted pause frame.

The CPGMAC_SL will transmit a pause frame to the reserved multicast address at the first available opportunity (immediately if currently idle, or following the completion of the frame currently being transmitted). The pause frame will contain the maximum possible value for the pause time (0xFFFF). The MAC will count the receive pause frame time (decrements 0xFF00 downto zero) and retransmit an outgoing pause frame if the count reaches zero. When the flow control request is removed, the MAC will transmit a pause frame with a zero pause time to cancel the pause request.

Note that transmitted pause frames are only a request to the other end station to stop transmitting. Frames that are received during the pause interval will be received normally (provided the Rx FIFO is not full).

Pause frames will be transmitted if enabled and triggered regardless of whether or not the port is observing the pause time period from an incoming pause frame.

The CPGMAC_SL will transmit pause frames as described below:

- The 48-bit reserved multicast destination address 01.80.C2.00.00.01.
- The 48-bit source address –**SL_SA(47:0)**.
- The 16-bit length/type field containing the value 88.08
- The 16-bit pause opcode equal to 00.01
- The 16-bit pause time value FF.FF. A pause-quantum is 512 bit-times. Pause frames sent to cancel a pause request will have a pause time value of 00.00.
- Zero padding to 64-byte data length (The CPGMAC_SL will transmit only 64 byte pause frames).
- The 32-bit frame-check sequence (CRC word).

All quantities above are hexadecimal and are transmitted most-significant byte first. The least-significant bit is transferred first in each byte.

If **rx_flow_en** is cleared to zero while the pause time is nonzero, then the pause time will be cleared to zero and a zero count pause frame will be sent.

13.3.1.9.2.2 Transmit Flow Control

Incoming pause frames are acted upon, when enabled, to prevent the CPGMAC_SL from transmitting any further frames. Incoming pause frames are only acted upon when the **fullduplex** and **tx_flow_en** bits in the **MacControl** register are set. Pause frames are not acted upon in half-duplex mode. Pause frame action will be taken if enabled, but normally the frame will be filtered and not transferred to memory. MAC control frames will be transferred to memory if the **rx_cmf_en** (Copy MAC Frames) bit in the **MacControl** register is set. The **tx_flow_en** and **fullduplex** bits effect whether or not MAC control frames are acted upon, but they have no effect upon whether or not MAC control frames are transferred to memory or filtered.

Pause frames are a subset of MAC Control Frames with an opcode field=0x0001. Incoming pause frames will only be acted upon by the port if:

Copyright © 2011, Texas Instruments Incorporated

- **tx_flow_en** is set in **MacControl**, and
- the frame's length is 64 to **rx_maxlen** bytes inclusive, and
- the frame contains no crc error or align/code errors.

The pause time value from valid frames will be extracted from the two bytes following the opcode. The pause time will be loaded into the port's transmit pause timer and the transmit pause time period will begin.

If a valid pause frame is received during the transmit pause time period of a previous transmit pause frame then:

- if the destination address is not equal to the reserved multicast address or any enabled or disabled unicast address, then the transmit pause timer will immediately expire, or
- if the new pause time value is zero then the transmit pause timer will immediately expire, else
- the port transmit pause timer will immediately be set to the new pause frame pause time value. (Any remaining pause time from the previous pause frame will be discarded).

If **tx_flow_en** in **MacControl** is cleared, then the pause-timer will immediately expire.

The port will not start the transmission of a new data frame any sooner than 512-bit times after a pause frame with a non-zero pause time has finished being received (**MRXDV** going inactive). No transmission will begin until the pause timer has expired (the port may transmit pause frames in order to initiate outgoing flow control). Any frame already in transmission when a pause frame is received will be completed and unaffected.

Incoming pause frames consist of the below:

- A 48-bit destination address equal to:
- The reserved multicast destination address 01.80.C2.00.00.01, or
- The SL_SA(47:0) input mac source address.
- The 48-bit source address of the transmitting device.
- The 16-bit length/type field containing the value 88.08
- The 16-bit pause opcode equal to 00.01
- The 16-bit pause_time. A pause-quantum is 512 bit-times.
- Padding to 64-byte data length.
- The 32-bit frame-check sequence (CRC word).

All quantities above are hexadecimal and are transmitted most-significant byte first. The least-significant bit is transferred first in each byte.

The padding is required to make up the frame to a minimum of 64 bytes. The standard allows pause frames longer than 64 bytes to be discarded or interpreted as valid pause frames. The CPGMAC_SL will recognize any pause frame between 64 bytes and **rx_maxlen** bytes in length.

13.3.1.10 Short Gap

The port 1 (and port 2) transmit inter-packet gap (IPG) may be shortened by eight bit times when enabled and triggered. The **tx_short_gap_en** bit in the **SL1_MacControl (SL2_MacControl)** register enables the gap to be shortened when triggered. The condition is triggered when the port 1 (port 2) transmit FIFO has a user defined number of FIFO blocks used. The port 1 transmit FIFO blocks used determines if the port 1 gap is shortened, and the port 2 transmit FIFO blocks used determines if the port 2 gap is shortened. The **CPSW_Gap_Thresh** register value determines the port 1 short gap threshold, and the **CPSW_Gap_Thresh** register value determines the port 2 short gap threshold.

13.3.1.11 Switch Latency

The CPSW_3G is a store and forward switch. The switch latency is defined as the amount of time between the end of packet reception of the received packet to the start of the output packet transmit.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Mode	Latency
Gig (1000)	880ns
100	1.3us
10	6.5us

13.3.1.12 Emulation Control

The emulation control input (EMUSUSP) and submodule emulation control registers allow CPSW_3G operation to be completely or partially suspended. There are three CPSW_3G submodules that contain emulation control registers (CPGMAC_SL1, CPGMAC_SL2, and CPDMA). The submodule emulation control registers must be accessed to facilitate CPSW_3G emulation control. The CPSW_3G module enters the emulation suspend state if all three submodules are configured for emulation suspend and the emulation suspend input is asserted. A partial emulation suspend state is entered if one or two submodules is configured for emulation suspend and the emulation suspend input is asserted. Emulation suspend occurs at packet boundaries. The emulation control feature is implemented for compatibility with other peripherals.

CPGMAC_SL Emulation Control

The emulation control input (**TBEMUSUP**) and register bits (**soft** and **free** in the **EMControl** register) allow CPGMAC_SL operation to be suspended. When the emulation suspend state is entered, the CPGMAC_SL will stop processing receive and transmit frames at the next frame boundary. Any frame currently in reception or transmission will be completed normally without suspension. For receive, frames that are detected by the CPGMAC_SL after the suspend state is entered are ignored. Emulation control is implemented for compatibility with other peripherals.

CPDMA Emulation Control

The emulation control input (**TBEMUSUP**) and register bits (**soft** and **free** in the **EMControl** register) allow CPDMA operation to be suspended. When the emulation suspend state is entered, the CPDMA will stop processing receive and transmit frames at the next frame boundary. Any frame currently in reception or transmission will be completed normally without suspension. For transmission, any complete or partial frame in the tx cell fifo will be transmitted. For receive, frames that are detected by the CPDMA after the suspend state is entered are ignored. No statistics will be kept for ignored frames. Emulation control is implemented for compatibility with other peripherals

The following table shows the operations of the emulation control input and register bits:

Table 13-17. Operations of Emulation Control Input and Register Bits

EMUSUSP	soft	free	Description
0	X	X	Normal Operation
1	0	0	Normal Operation
1	1	0	Emulation Suspend
1	X	1	Normal Operation

13.3.1.13 Software IDLE

The submodule software idle register bits enable CPSW_3G operation to be completely or partially suspended by software control. There are three CPSW_3G submodules that contain software idle register bits (CPGMAC_SL1, CPGMAC_SL2, and CPDMA). Each of the three submodules may be individually commanded to enter the idle state. The idle state is entered at packet boundaries, and no further packet operations will occur on an idled submodule until the idle command is removed. The CPSW_3G module enters the idle state when all three submodules are commanded to enter and have entered the idle state. Idle status is determined by reading or polling the three submodule idle bits. The CPSW_3G is in the idle state when all three submodules are in the idle state. The **CPSW_Soft_Idle** bit may be set if desired after the submodules are in the idle state. The **CPSW_Soft_Idle** bit causes packets to not be transferred from one FIFO to another FIFO internal to the switch.

Copyright © 2011, Texas Instruments Incorporated

13.3.1.14 Software Reset

The CPSW_3G software reset register, CPSW_3GSS software reset register and the three submodule software reset registers enable the CPSW_3GSS to be reset by software. There are three CPSW_3G submodules that contain software reset registers (CPGMAC_SL1, CPGMAC_SL2, and CPDMA). Each of the three submodules may be individually commanded to be reset by software.

For the CPDMA, the reset state is entered at packet boundaries, at which time the CPDMA reset occurs. The CPGMAC_SL soft reset is immediate. Submodule reset status is determined by reading or polling the submodule reset bit. If the submodule reset bit is read as a one, then the reset process has not yet completed. The submodule soft reset process could take up to 2ms each. The reset has completed if the submodule reset bit is read as a zero.

After all three submodules (in any order) have been reset and a read of each submodule reset bit indicates that the reset process is complete, the CPSW_3G software reset register bit may be written to complete the CPSW_3G module software reset operation. The CPSW_3G software reset bit controls the reset of the FIFO's, the statistics submodule, and the address lookup engine (ALE). The CPSW_3G software reset is immediate and will be indicated by reading a zero from the soft reset bit.

The CPSW_3GSS software reset bit controls the reset of the INT, REGS and CPPI. The CPSW_3GSS software reset is immediate and will be indicated by reading a zero from the soft reset bit.

13.3.1.15 FIFO Loopback

FIFO loopback mode is entered when the **fifo_loopback** bit in the **CPSW_Control** register is set. FIFO loopback mode causes packets received on a port to be turned around and transmitted back on the same port. Port 0 receive is fixed on channel zero in FIFO loopback mode. The **RXSOFOVERRUN** statistic is incremented for each packet sent in FIFO loopback mode. Packets sent in with errors are returned with errors (they are not dropped). FIFO loopback is intended as a simple mechanism for test purposes. FIFO loopback should be performed in fullduplex mode only.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.3.2 Common Platform Time Sync

The Common Platform Time Sync (CPTS) module is used to facilitate host control of time sync operations. It enables compliance with the IEEE 1588-2008 standard for a precision clock synchronization protocol.

13.3.2.1 Architecture

Figure 13-8. CPTS Block Diagram

Figure 1 shows the architecture of the CPTS module inside the 3PSW Ethernet Subsystem. Time stamp values for every packet transmitted or received on either port of the 3PSW are recorded. At the same time, each packet is decoded to determine if it is a valid time sync event. If so, an event is loaded into the Event FIFO for processing containing the recorded time stamp value when the packet was transmitted or received.

In addition, both hardware (**HWx_TS_PUSH**) and software (**TS_PUSH**) can be used to read the current time stamp value though the Event FIFO

The reference clock used for the time stamp (**RCLK**) is sourced from one of the five sources, as shown in Figure 2. The PLL used for sourcing **RCLK** can be selected by configuring the **RMII _REFCLK_SRC** register in the Device Control Module. Please refer to the Device Control Module document for more details. Note that if RGMII mode is used, RCLK must be exactly 250 MHz.

Figure 13-9. CPTS RCLK Source

13.3.2.2 Common Platform Time Sync Initialization

The CPTS module should be configured using the following steps:

- Reset the CPTS module
- Clear the CPTS_EN bit in the CONTROL register
- Write the CPTS_RFT_CLK value in the RMII_RFTCLK_SEL register with the desired reference clock selection
- Set the CPTS_EN bit in the CONTROL register
- If using interrupts and not polling, enable the interrupt by setting the TS_PEND_EN bit in the TS_INT_ENABLE register.

13.3.2.3 Time Stamp Value

The time stamp value is a 32-bit value that increments on each RCLK rising edge when CPTS_EN is set to one. When CPTS_EN is cleared to zero the time stamp value is reset to zero. If more than 32-bits of time stamp are required by the application, the host software must maintain the necessary number of upper bits. The upper time stamp value should be incremented by the host when the rollover event is detected.

For test purposes, the time stamp can be written via the time stamp load function (TS_LOAD_VAL and TS_LOAD_EN registers).

13.3.2.4 Event FIFO

All time sync events are push onto the Event FIFO. There are 16 locations in the event FIFO with no overrun indication supported. Software must service the event FIFO in a timely manner to prevent FIFO overrun.

13.3.2.5 Time Sync Events

Time Sync events are 64-bit values that are pushed onto the event FIFO and read in two 32-bit reads. The definition for the two 32-bit registers, EVENT_HIGH and EVENT_LOW, is shown in the Register section of this document.

There are five types of sync events

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

- Time stamp push event
- Time stamp counter rollover event
- Time stamp counter half-rollover event
- Ethernet receive event
- Ethernet transmit event

13.3.2.5.1 Time Stamp Push Event

Software can obtain the current time stamp value (at the time of the write) by initiating a time stamp push event. The push event is initiated by setting the TS_PUSH bit of the TS_PUSH register. The time stamp value is returned in the event, along with a time stamp push event code. Software should not push a second time stamp event on to the FIFO until the first time stamp value has been read from the event FIFO.

13.3.2.5.2 Time Stamp Counter Rollover Event

The CPTS module contains a 32-bit time stamp value. The counter upper bits are maintained by host software. The rollover event indicates to software that the time stamp counter has rolled over from 0xFFFF_FFFF to 0x0000_0000, and the software maintained upper count value should be incremented.

13.3.2.5.3 Time Stamp Counter Half-rollover Event

The CPTS includes a time stamp counter half-rollover event. The half-rollover event indicates to software that the time stamp value has incremented from 0x7FFF_FFFF to 0x8000_0000. The half-rollover event is included to enable software to correct a misaligned event condition.The half-rollover event is included to enable software to determine the correct time for each event that contains a valid time stamp value – such as an Ethernet event. If an Ethernet event occurs around a counter rollover (full rollover), the rollover event could possibly be loaded into the event FIFO before the Ethernet event, even though the Ethernet event time was actually taken before the rollover. Figure 3 below shows a misalignment condition.

Figure 13-10. Event FIFO Misalignment Condition

Host software must detect and correct for misaligned event conditions. For every event after a rollover and before a half-rollover, software must examine the time stamp most significant bit. If bit 31 of the time stamp value is low (0x0000_0000 through 0x7FFF_FFFF), then the event time stamp was taken after the rollover and no correction is required. If the value is high (0x8000_0000 through 0xFFFF_FFFF), the time stamp value was taken before the rollover and a misalignment is detected. The misaligned case indicates to software that it must subtract one from the upper count value stored in software to calculate the correct time for the misaligned event.The misaligned event occurs only on the rollover boundary and not on the half-rollover boundary. Software only needs to check for misalignment from a rollover event to a half-rollover event.

13.3.2.5.4 Hardware Time Stamp Push Event

There are four hardware time stamp inputs (HW1/4_TS_PUSH) that can cause hardware time stamp push events to be loaded into the Event FIFO. Each hardware time stamp input is internally connected to the PORTIMERPWM output of each timer as shown in Figure 4.

Figure 13-11. HW1/4_TSP_PUSH Connection

The event is loaded into the event FIFO on the rising edge of the timer, and the PORT_NUMBER field in the EVENT_HIGH register indicates the hardware time stamp input that caused the event.

Each hardware time stamp input must be asserted for at least 10 periods of the selected RCLK clock. Each input can be enabled or disabled by setting the respective bits in the ***CONTROL*** register.

Hardware time stamps are intended to be an extremely low frequency signals, such that the event FIFO does not overrun. Software must keep up with the event FIFO and ensure that there is no overrun, or events will be lost.

Copyright © 2011, Texas Instruments Incorporated

13.3.2.5.5 Ethernet Port Events

Packets transmitted or received on each Ethernet port can generate Ethernet Transmit Events or Ethernet Receive Events, respectively. The CPTS hardware will decode each packet to determine if it is a valid CPTS time sync event.

According to the IEEE 802.3 Ethernet standard, each Ethernet frame contains a 2-octet EtherType field to indicate which protocol is encapsulated in the PayLoad field, as shown in Figure 5(1). For standard time sync packets, this will contain the EtherType for the Precision Time Protocol (IEEE 1588), which is defined as 0x88F7. The CPTS hardware will compare this field to the **pX_ts_ltype** field in the **PX_TS_CTL** register, which should also be programmed to 0x88F7.

When a virtual LAN is used, an addition 4-octet 802.1Q tag is inserted in the Ethernet frame before the EtherType field, as shown in Figure 5(2). To indicate to the CPTS hardware that a virtual LAN is in use, the **pX_ts_tx_vlan_ltype1_en/pX_ts_rx_vlan_ltype1_en (**or **pX_ts_tx_vlan_ltype2_en/pX_ts_rx_vlan_ltype2_en)** enable bit must be set in the **PX_TS_CTL** register. The EtherType for the 802.1Q tag is defined as 0x8100, and the CTPS hardware will compare this value to the **pX_ts_vlan_ltype1** (or **pX_ts_vlan_ltype2** depending on which enable bit was set**)** field in the **PX_TS_VLAN_LTYPE** register, which should also be programmed to 0x8100.

When two stacked VLANs are used, two additional 4-octet 801.Q tags are inserted in the Ethernet frame before the EtherType field, as shown in Figure 5(3). In this case, both **pX_ts_vlan_ltype1** and **pX_ts_vlan_ltype2** must be enabled. The outer tag must match the value of the **pX_ts_vlan_ltype1** field, and the inner tag must match the value of the **pX_ts_vlan_ltype2** field.

Figure 13-12. Partial Ethernet-II frames showing register mapping of EtherTypes for a simple frame (1), a single 1Q tag added (2), and two 1Q tags added (3).

To enable transmit/receive event packets on a given Ethernet port, perform the following steps, where **X** is the port number:

- Set the **pX_ts_tx_en/pX_ts_rx_en** bit in the **PX_TS_CTL** register to enable transmit/receive event packets
- Configure the **PX_TS_SEQ_LTYPE** register:
  - Set the **pX_ts_ltype** field to 0x88F7, which corresponds to the Precision Time Protocol (IEEE 1588) EtherType.
  - Set the **pX_ts_seq_id_offset** field to 0x1E (d30), which is the sequence ID offset in the common message header given in the IEEE 1588 specification.
- To enable support for VLAN tagging (IEEE 802.1Q):
  - Set the **pX_ts_tx_vlan_ltype1_en/pX_ts_rx_vlan_ltype1_en** or **pX_ts_tx_vlan_ltype2_en/pX_ts_rx_vlan_ltype2_en** bit in the **PX_TS_CTL** register.
  - Set the **pX_ts_vlan_ltype1** or **pX_ts_vlan_ltype2** field (matching what was used in 3a) of the **PX_TS_VLAN_LTYPE** to 0x8100, which corresponds to the VLAN-tagged frame (IEEE 802.1Q) EtherType.
- To enable support for up to two stacked VLANs (IEEE 802.1ad):
  - Set the **pX_ts_tx_vlan_ltype1_en/pX_ts_rx_vlan_ltype1_en** and **pX_ts_tx_vlan_ltype2_en/pX_ts_rx_vlan_ltype2_en** bit in the **PX_TS_CTL** register.
  - Set the **pX_ts_vlan_ltype1** field of the **PX_TS_VLAN_LTYPE** register to match the EtherType of the outer tag.
  - Set the **pX_ts_vlan_ltype2** field of the **PX_TS_VLAN_LTYPE** register to 0x8100, which corresponds to the VLAN-tagged frame (IEEE 802.1Q) EtherType of the inner tag.
- Set the **pX_ts_msg_type_en** field of the **PX_TS_CTL** register to choose which types of time stamp messages will push events onto the Event FIFO. The following table enumerates the message types defined in the IEEE 1588-2008 specification.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 13-18. Values of messageType field

Message Type	Value (hex)
Sync	0
Delay_Req	1
Pdelay_Req	2
Pdelay_Resp	3
Reserved	4-7
Follow_Up	8
Delay_Resp	9
Pdelay_Resp_Follow_Up	A
Announce	B
Signaling	C
Management	D
Reserved	E-F

Once a transmitted or received packet is determined to be a valid time sync packet, the Ethernet Transmit Event or Ethernet Receive Event is loaded onto the Event FIFO. The **EVENT_HIGH** register contains the Message Type and Sequence ID values from the original time sync packet. The **EVENT_LOW** register contains the time stamp value when the packet arrived at the corresponding port.

13.3.2.6 Interrupt Handling

When an event is push onto the Event FIFO, an interrupt can be generated to indicate to software that a time sync event occurred. The following steps should be taken to process time sync events using interrupts:

- Enable the TS_PEND interrupt by setting the **TS_PEND_EN** bit of the **TS_Int_enable** register.
- Upon interrupt, read the **Event_Low** and **Event_High** register values.
- Set the EVENT_POP field (bit zero) of the **Event_Pop** register to pop the previously read value off of the event FIFO.
- Process the interrupt as required by the application software

Software has the option of processing more than a single event from the event FIFO in the interrupt

service routine in the following way:

- Enable the **TS_PEND** interrupt by setting the **TS_PEND_EN** bit of the **TS_Int_enable** register.
- Upon interrupt, read the **Event_Low** and **Event_High** register values.
- Set the **EVENT_POP** bit of the **Event_Pop** register to pop the previously read value off of the event FIFO.
- Wait for an amount of time greater than four **RCLK** periods plus four **VBUSP_CLK** periods.
- Read the **ts_pend_raw** bit in the **TS_Intstat_raw** register to determine if another valid event is in the event FIFO. If it is asserted then goto step 3. Otherwise goto step 6.
- Process the interrupt(s) as required by the application software

Software also has the option of disabling the interrupt and polling the **ts_pend_raw** bit of the **TS_Intstat_raw** register to determine if a valid event is on the event FIFO.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.3.3 CPPI Buffer Descriptors

The buffer descriptor is a central part of the 3PSW Ethernet Subsytem and is how the application software describes Ethernet packets to be sent and empty buffers to be filled with incoming packet data.

Host Software sends and receives network frames via the CPPI 3.0 compliant host interface. The host interface includes module registers and host memory data structures. The host memory data structures are buffer descriptors and data buffers. Buffer descriptors are data structures that contain information about a single data buffer. Buffer descriptors may be linked together to describe frames or queues of frames for transmission of data and free buffer queues available for received data.

Note: The 8k bytes of Ethernet Subsystem CPPI RAM begin at address 0x4a102000 and end at 0x4a103FFF from the 3PSW perspective. The buffer descriptors programmed to access the CPPI RAM memory should use the address range from 0x4a102000.

13.3.3.1 TX Buffer Descriptors

A TX buffer descriptor is a contiguous block of four 32-bit data words aligned on a 32-bit word boundary.

13.3.3.1.1 Cppi Tx Data Word - 0

Next Descriptor Pointer

The next descriptor pointer points to the 32-bit word aligned memory address of the next buffer descriptor in the transmit queue. This pointer is used to create a linked list of buffer descriptors. If the value of this pointer is zero, then the current buffer is the last buffer in the queue. The software application must set this value prior to adding the descriptor to the active transmit list. This pointer is not altered by the EMAC.The value of pNext should never be altered once the descriptor is in an active transmit queue, unless its current value is NULL. If the pNext pointer is initially NULL, and more packets need to be queued for transmit, the software application may alter this pointer to point to a newly appended descriptor. The EMAC will use the new pointer value and proceed to the next descriptor unless the pNext value has already been read. In this latter case, the transmitter will halt on the transmit channel in question, and the software application may restart it at that time. The software can detect this case by checking for an end of queue (EOQ) condition flag on the updated packet descriptor when it is returned by the EMAC.

13.3.3.1.2 Cppi Tx Data Word - 1

Buffer Pointer

The byte aligned memory address of the buffer associated with the buffer descriptor. The host sets the **buffer_pointer**. The software application must set this value prior to adding the descriptor to the active transmit list. This pointer is not altered by the EMAC.

13.3.3.1.3 Cppi Tx Data Word - 2

Buffer _Offset

Buffer Offset – Indicates how many unused bytes are at the start of the buffer. A value of 0x0000 indicates that no unused bytes are at the start of the buffer and that valid data begins on the first byte of the buffer. A value of 0x000F (decimal 15) indicates that the first 15 bytes of the buffer are to be ignored by the port and that valid buffer data starts on byte 16 of the buffer. The host sets the buffer_offset value (which may be zero to the buffer length minus 1). Valid only on sop.

Buffer _Length

Buffer Length – Indicates how many valid data bytes are in the buffer. Unused or protocol specific bytes at the beginning of the buffer are not counted in the Buffer Length field. The host sets the buffer_length. The buffer_length must be greater than zero.

13.3.3.1.4 Cppi Tx Data Word – 3

Packet Length

Copyright © 2011, Texas Instruments Incorporated

Specifies the number of bytes in the entire packet. Offset bytes are not included. The sum of the buffer_length fields should equal the packet_length. Valid only on SOP. The packet length must be greater than zero. The packet data will be truncated to the packet length if the packet length is shorter than the sum of the packet buffer descriptor buffer lengths. A host error occurs if the packet length is greater than the sum of the packet buffer descriptor buffer lengths

Start of Packet (SOP) Flag

When set, this flag indicates that the descriptor points to a packet buffer that is the start of a new packet.In the case of a single fragment packet, both the SOP and end of packet (EOP) flags are set. Otherwise,the descriptor pointing to the last packet buffer for the packet sets the EOP flag. This bit is set by the software application and is not altered by the EMAC.

0 - Not start of packet buffer

1 - Start of packet buffer

End of Packet (EOP) Flag

When set, this flag indicates that the descriptor points to a packet buffer that is last for a given packet. In the case of a single fragment packet, both the start of packet (SOP) and EOP flags are set. Otherwise, the descriptor pointing to the last packet buffer for the packet sets the EOP flag. This bit is set by the software application and is not altered by the EMAC.

0 - Not end of packet buffer.

1 - End of packet buffer.

Ownership

When set this flag indicates that all the descriptors for the given packet (from SOP to EOP) are currently owned by the EMAC. This flag is set by the software application on the SOP packet descriptor before adding the descriptor to the transmit descriptor queue. For a single fragment packet, the SOP, EOP, and OWNER flags are all set. The OWNER flag is cleared by the EMAC once it is finished with all the descriptors for the given packet. Note that this flag is valid on SOP descriptors only.

0 - The packet is owned by the host

1 - The packet is owned by the port

EOQ

When set, this flag indicates that the descriptor in question was the last descriptor in the transmit queue for a given transmit channel, and that the transmitter has halted. This flag is initially cleared by the software application prior to adding the descriptor to the transmit queue. This bit is set by the EMAC when the EMAC identifies that a descriptor is the last for a given packet (the EOP flag is set), and there are no more descriptors in the transmit list (next descriptor pointer is NULL).

The software application can use this bit to detect when the EMAC transmitter for the corresponding channel has halted. This is useful when the application appends additional packet descriptors to a transmit queue list that is already owned by the EMAC. Note that this flag is valid on EOP descriptors only.

0 - The Tx queue has more packets to transfer.

1 - The Descriptor buffer is the last buffer in the last packet in the queue.

teardown_Complete

This flag is used when a transmit queue is being torn down, or aborted, instead of allowing it to be

transmitted. This would happen under device driver reset or shutdown conditions. The EMAC sets this bit in the SOP descriptor of each packet as it is aborted from transmission. Note that this flag is valid on SOP descriptors only. Also note that only the first packet in an unsent list has the TDOWNCMPLT flag set. Subsequent descriptors are not processed by the EMAC.

0 - The port has not completed the teardown process.

1 - The port has completed the commanded teardown process.

pass_crc

Copyright © 2011, Texas Instruments Incorporated

This flag is set by the software application in the SOP packet descriptor before it adds the descriptor to the transmit queue. Setting this bit indicates to the EMAC that the 4 byte Ethernet CRC is already present in the packet data, and that the EMAC should not generate its own version of the CRC.When the CRC flag is cleared, the EMAC generates and appends the 4-byte CRC. The buffer length and packet length fields do not include the CRC bytes. When the CRC flag is set, the 4-byte CRC is supplied by the software application and is already appended to the end of the packet data. The buffer length and packet length fields include the CRC bytes, as they are part of the valid packet data. Note that this flag is valid on SOP descriptors only.

0 – The CRC is not included with the packet data and packet length.

1 – The CRC is included with the packet data and packet length.

to_port

To Port – Port number to send the directed packet to. This field is set by the host. This field is valid on SOP. Directed packets go to the directed port, but an ALE lookup is performed to determine untagged egress in VLAN_AWARE mode.

1 – Send the packet to port 1 if to_port_en is asserted.

2 – Send the packet to port 2 if to_port_en is asserted.

To_port_enable

To Port Enable – Indicates when set that the packet is a directed packet to be sent to the to_port field port number. This field is set by the host. The packet is sent to one port only (index not mask). This bit is valid on SOP.

0 – not a directed packet

1 – directed packet

13.3.3.2 RX Buffer Descriptors

A RX buffer descriptor is a contiguous block of four 32-bit data words aligned on a 32-bit word boundary.

13.3.3.2.1 Cppi Rx Data Word – 0

next_descriptor_pointer

The 32-bit word aligned memory address of the next buffer descriptor in the RX queue. This is the mechanism used to reference the next buffer descriptor from the current buffer descriptor. If the value of this pointer is zero then the current buffer is the last buffer in the queue. The host sets the **next_descriptor_pointer**.

13.3.3.2.2 Cppi Rx Data Word – 1

buffer_pointer

The byte aligned memory address of the buffer associated with the buffer descriptor. The host sets the **buffer_pointer**.

13.3.3.2.3 Cppi Rx Data Word – 2

Buffer _Offset

Buffer Offset – Indicates how many unused bytes are at the start of the buffer. A value of 0x0000 indicates that there are no unused bytes at the start of the buffer and that valid data begins on the first byte of the buffer. A value of 0x000F (decimal 15) indicates that the first 15 bytes of the buffer are to be ignored by the port and that valid buffer data starts on byte 16 of the buffer. The port writes the **buffer_offset** with the value from the **rx_buffer_offset** register value. The host initializes the **buffer_offset** to zero for free buffers. The **buffer_length** must be greater than the **rx_buffer_offset** value. The buffer offset is valid only on **sop**.

Buffer _Length

Copyright © 2011, Texas Instruments Incorporated

Buffer Length – Indicates how many valid data bytes are in the buffer. Unused or protocol specific bytes at the beginning of the buffer are not counted in the Buffer Length field. The host initializes the **buffer_length**, but the port may overwrite the host initiated value with the actual buffer length value on SOP and/or EOP buffer descriptors. SOP buffer length values will be overwritten if the packet size is less than the size of the buffer or if the offset is nonzero. EOP buffer length values will be overwritten if the entire buffer is not filled up with data. The **buffer_length** must be greater than zero.

13.3.3.2.4 Cppi Rx Data Word – 3

Packet Length

Specifies the number of bytes in the entire packet. The packet length is reduced to 12-bits. Offset bytes are not included. The sum of the buffer_length fields should equal the packet_length. Valid only on SOP.

Start of Packet (SOP) Flag

When set, this flag indicates that the descriptor points to a packet buffer that is the start of a new packet.In the case of a single fragment packet, both the SOP and end of packet (EOP) flags are set. Otherwise, the descriptor pointing to the last packet buffer for the packet has the EOP flag set. This flag is initially cleared by the software application before adding the descriptor to the receive queue. This bit is set by the EMAC on SOP descriptors.

End of Packet (EOP) Flag

When set, this flag indicates that the descriptor points to a packet buffer that is last for a given packet. In the case of a single fragment packet, both the start of packet (SOP) and EOP flags are set. Otherwise, the descriptor pointing to the last packet buffer for the packet has the EOP flag set. This flag is initially cleared by the software application before adding the descriptor to the receive queue. This bit is set by the EMAC on EOP descriptors.

Ownership (OWNER) Flag

When set, this flag indicates that the descriptor is currently owned by the EMAC. This flag is set by the software application before adding the descriptor to the receive descriptor queue. This flag is cleared by the EMAC once it is finished with a given set of descriptors, associated with a received packet. The flag is updated by the EMAC on SOP descriptor only. So when the application identifies that the OWNER flag is cleared on an SOP descriptor, it may assume that all descriptors up to and including the first with the EOP flag set have been released by the EMAC. (Note that in the case of single buffer packets, the same descriptor will have both the SOP and EOP flags set.)

End of Queue (EOQ) Flag

When set, this flag indicates that the descriptor in question was the last descriptor in the receive queue for a given receive channel, and that the corresponding receiver channel has halted. This flag is initially cleared by the software application prior to adding the descriptor to the receive queue. This bit is set by the EMAC when the EMAC identifies that a descriptor is the last for a given packet received (also sets the EOP flag), and there are no more descriptors in the receive list (next descriptor pointer is NULL).The software application can use this bit to detect when the EMAC receiver for the corresponding channel has halted. This is useful when the application appends additional free buffer descriptors to an active receive queue. Note that this flag is valid on EOP descriptors only.

Teardown Complete (TDOWNCMPLT) Flag

This flag is used when a receive queue is being torn down, or aborted, instead of being filled with received data. This would happen under device driver reset or shutdown conditions. The EMAC sets this bit in the descriptor of the first free buffer when the tear down occurs. No additional queue processing is performed.

Pass CRC (PASSCRC) Flag

This flag is set by the EMAC in the SOP buffer descriptor if the received packet includes the 4-byte CRC.This flag should be cleared by the software application before submitting the descriptor to the receive queue.

Long (Jabber) Flag

Copyright © 2011, Texas Instruments Incorporated

This flag is set by the EMAC in the SOP buffer descriptor, if the received packet is a jabber frame and was not discarded because the RX_CEF_EN bit was set in the MacControl. Jabber frames are frames that exceed the RXMAXLEN in length, and have CRC, code, or alignment errors.

Short (Fragment) Flag

This flag is set by the EMAC in the SOP buffer descriptor, if the received packet is only a packet fragment and was not discarded because the RX_CSF_EN bit was set in the MacControl.

Control Flag

This flag is set by the EMAC in the SOP buffer descriptor, if the received packet is an EMAC control frame and was not discarded because the RX_CMF_EN bit was set in the MacControl.

Overrun Flag

This flag is set by the EMAC in the SOP buffer descriptor, if the received packet was aborted due to a receive overrun.

Pkt_error Flag

Packet Contained Error on Ingress –

00 – no error

01 – CRC error on ingress

10 – Code error on ingress

11 – Align error on ingress

rx_vlan_encap

VLAN Encapsulated Packet – Indicates when set that the packet data contains a 32-bit VLAN header word that is included in the packet byte count. This field is set by the port to be the value of the CPSW control register **rx_vlan_encap** bit

from_port

From Port – Indicates the port number that the packet was received on (ingress to the switch).

13.3.4 MDIO

The MII Management I/F module implements the 802.3 serial management interface to interrogate and control two Ethernet PHYs simultaneously using a shared two-wire bus. Two user access registers to control and monitor up to two PHYs simultaneously.

13.3.4.1 MII Management Interface Frame Formats

The following tables show the read and write format of the 32-bit MII Management interface frames, respectively.

Table 13-19. MDIO Read Frame Format

Preamble	Start Delimiter	Operation Code	PHY Address	Register Address	Turnaround	Data
0xFFFFFFFF	01	10	AAAAA	RRRRR	Z0	DDDD.DDDD.DDDD.DDDD

Table 13-20. MDIO Write Frame Format

Preamble	Start Delimiter	Operation Code	PHY Address	Register Address	Turnaround	Data
0xFFFFFFFF	01	00	AAAAA	RRRRR	10	DDDD.DDDD.DDDD.DDDD

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

The default or idle state of the two wire serial interface is a logic one. All tri-state drivers should be disabled and the PHY's pull-up resistor will pull the **MDIO** line to a logic one. Prior to initiating any other transaction, the station management entity shall send a preamble sequence of 32 contiguous logic one bits on the **MDIO** line with 32 corresponding cycles on **MDCLK** to provide the PHY with a pattern that it can use to establish synchronization. A PHY shall observe a sequence of 32 contiguous logic one bits on **MDIO** with 32 corresponding **MDCLK** cycles before it responds to any other transaction.

Preamble

The start of a frame is indicated by a preamble, which consists of a sequence of 32 contiguous bits all of which are a "1". This sequence provides the PHY a pattern to use to establish synchronization.

Start Delimiter

The preamble is followed by the start delimiter which is indicated by a "01" pattern. The pattern assures transitions from the default logic one state to zero and back to one.

Operation Code

The operation code for a read is "10", while the operation code for a write is a "00".

PHY Address

The PHY address is 5 bits allowing 32 unique values. The first bit transmitted is the MSB of the PHY address.

Register Address

The Register address is 5 bits allowing 32 registers to be addressed within each PHY. Refer to the 10/100 PHY address map for addresses of individual registers.

Turnaround

An idle bit time during which no device actively drives the MDIO signal shall be inserted between the register address field and the data field of a read frame in order to avoid contention. During a read frame, the PHY shall drive a zero bit onto MDIO for the first bit time following the idle bit and preceding the Data field. During a write frame, this field shall consist of a one bit followed by a zero bit.

Data

The Data field is 16 bits. The first bit transmitted and received is the MSB of the data word.

13.3.4.2 Functional description

The MII Management I/F will remain idle until enabled by setting the **enable** bit in the **MDIOControl** register. The MII Management I/F will then continuously poll the link status from within the Generic Status Register of all possible 32 PHY addresses in turn recording the results in the **MDIOLink** register. The **linksel** bit in the **MDIOUserPhySel** register determines the status input that is used. A change in the link status of the two PHYs being monitored will set the appropriate bit in the **MDIOLinkIntRaw** register and the **MDIOLinkIntMasked** register, if enabled by the **linkint_enable** bit in the **MDIOUserPhySel** register.

The **MDIOAlive** register is updated by the MII Management I/F module if the PHY acknowledged the read of the generic status register. In addition, any PHY register read transactions initiated by the host also cause the **MDIOAlive** register to be updated.

At any time, the host can define a transaction for the MII Management interface module to undertake using the **data, phyadr, regadr,** and **write** fields in a **MDIOUserAccess** register. When the host sets the **go** bit in this register, the MII Management interface module will begin the transaction without any further intervention from the host. Upon completion, the MII Management interface will clear the **go** bit and set the **userintraw** bit in the **MDIOUserIntRaw** register corresponding to the **MDIOUserAccess** register being used. The corresponding bit in the **MDIOUserIntMasked** register may also be set depending on the mask setting in the **MDIOUserIntMaskSet** and **MDIOUserIntMaskClr** registers. A round-robin arbitration scheme is used to schedule transactions which may queued by the host in different **MDIOUserAccess** registers. The host should check the status of the **go** bit in the **MDIOUserAccess** register before initiating a new transaction to ensure that the previous transaction has completed. The host can use the **ack** bit in the **MDIOUserAccess** register to determine the status of a read transaction.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

It is necessary for software to use the MII Management interface module to setup the auto-negotiation parameters of each PHY attached to a MAC port, retrieve the negotiation results, and setup the **MACControl** register in the corresponding MAC.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.3.5 Interrupts

The 3Port Switch Ethernet Subsystem generates 4 interrupt events.

13.3.5.1 Receive Packet Completion Pulse Interrupt (RX_PULSE)

The RX_PULSE interrupt is a pulse interrupt selected from the 3PSW RX_PEND [7:0] interrupts. The receive DMA controller has eight channels with each channel having a corresponding (RX_PEND[7:0]).

The following steps will enable the receive packet completion interrupt.

- Enable the required channel interrupts of the DMA engine by setting 1 to the appropriate bit in the RX_INTMASK_SET register.
- The receive completion interrupt(s) to be routed to RX_PULSE is selected by setting one or more bits in the receive interrupt enable register RX_EN . The masked interrupt status can be read in the address location of RX_STAT bit in the receive status register.

When the 3PSW completes a packet reception, the subsystem issues an interrupt to the CPU by writing the packet's last buffer descriptor address to the appropriate channel queue's receive completion pointer located in the state RAM block. The interrupt is generated by the write when enabled by the interrupt mask, regardless of the value written.

Upon interrupt reception, the CPU processes one or more packets from the buffer chain and then

acknowledges one or more interrupt(s) by writing the address of the last buffer descriptor processed to the queue's associated receive completion pointer (RX*n*_CP) in the receive DMA state RAM.

Upon reception of an interrupt, software should perform the following:

- Read the RX_STAT bit address location to determine which channel(s) caused the interrupt.
- Process received packets for the interrupting channel(s).
- Write the 3PSW completion pointer(s) (RX*n*_CP). The data written by the host (buffer descriptor address of the last processed buffer) is compared to the data in the register written by the subsystem (address of last buffer descriptor used by the subsystem). If the two values are not equal (which means that the 3PSW has received more packets than the CPU has processed), the receive packet completion interrupt signal remains asserted. If the two values are equal (which means that the host has processed all packets that the system has received), the pending interrupt is de-asserted. The value that the 3PSW is expecting is found by reading the receive channel*n* completion pointer register (RX*n*_CP).
- Write the value 1h to the CPDMA_EOI_VECTOR register.

To disable the interrupt:

- The eight channel interrupts may be individually disabled by writing to 1 the appropriate bit in the RX_INTMASK_CLEAR
- The receive completion pulse interrupt could be disabled by clearing to 0 all the bits of the RX_EN.

The software could still poll for the RX_INTSTAT_RAW and RX_INTSTAT_MASKED registers if the corresponding interrupts are enabled.

13.3.5.2 Transmit Packet Completion Pulse Interrupt (TX_PULSE)

The TX_PULSE interrupt is a pulse interrupt selected from the 3PSW TX_PEND [7:0] interrupts. The transmit DMA controller has eight channels with each channel having a corresponding (TX_PEND[7:0]).

To enable the transmit packet completion interrupt:

- Enable the required channel interrupts of the DMA engine by setting 1 to the appropriate bit in the TX_INTMASK_SET register.
- The transmit completion interrupt(s) to be routed to TX_PULSE is selected by setting one or more bits in the transmit interrupt enable register TX_EN .The masked interrupt status can be read in the address location of TX_STAT bit in the transmit status register.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

When the 3PSW completes the transmission of a packet, the 3PSW subsystem issues an interrupt to the CPU by writing the packet's last buffer descriptor address to the appropriate channel queue's transmit completion pointer located in the state RAM block. The interrupt is generated by the write when enabled by the interrupt mask, regardless of the value written.

Upon receiving an interrupt, software should perform the following:

- Read the TX_STAT bit address location to determine which channel(s) caused the interrupt
- Process received packets for the interrupting channel(s).
- Write the 3PSW completion pointer(s) (TX*n*_CP). The data written by the host (buffer descriptor address of the last processed buffer) is compared to the data in the register written by the 3PSW (address of last buffer descriptor used by the 3PSW). If the two values are not equal (which means that the 3PSW has transmitted more packets than the CPU has processed), the transmit packet completion interrupt signal remains asserted. If the two values are equal (which means that the host has processed all packets that the subsystem has transferred), the pending interrupt is cleared. The value that the 3PSW is expecting is found by reading the transmit channel*n* completion pointer register (TX*n*_CP).
- Write the 2h to the CPDMA_EOI_VECTOR register.

To disable the interrupt:

- The eight channel interrupts may be individually disabled by writing to 1 the appropriate bit in the TX_INTMASK_CLEAR.
- The receive completion pulse interrupt could be disabled by clearing to 0 all the bits of the TX_EN. The software could still poll for the TX_INTSTAT_RAW and TX_INTSTAT_MASKED registers if the corresponding interrupts are enabled.

13.3.5.3 Receive Threshold Pulse Interrupt (RX_THRESH_PULSE)

The RX_THRESH_PULSE interrupt is an immediate (non-paced) pulse interrupt selected from the CPSW_3G RX_THRESH_PEND[7:0] interrupts. The receive DMA controller has eight channels with each channel having a corresponding threshold pulse interrupt (RX_THRESH_PEND [7:0]).

To enable the receive threshold pulse Interrupt:

- Enable the required channel interrupts of the DMA engine by setting 1 to the appropriate bit in the RX_INTMASK_SET register.
- The receive threshold interrupt(s) to be routed to RX_THRESH_PULSE is selected by setting one or more bits in the receive threshold interrupt enable register RX_THRESH_EN. The masked interrupt status can be read in the address location of RX_THRESH_STAT bit in the receive threshold status register.

The RX_THRESH_PULSE is asserted when enabled when the channel's associated free buffer count RX*n*_FREEBUFFER is less than or equal to the corresponding RX*n*_PENDTHRESH register.

Upon receiving an interrupt, software should perform the following:

- Read the RX_THRESH_STAT bit address location to determine which channel(s) caused the interrupt.
- Process the received packets in order to add more buffers to any channel that is below the threshold value.
- Write the CPSW_3G completion pointer(s).
- Write the value 0h to the CPDMA_EOI_VECTOR register.

The threshold pulse interrupt is an immediate interrupt intended to indicate that software should immediately process packets to preclude an overrun condition from occurring for the particular channels.

To disable the interrupt:

- The eight channel receive threshold interrupts may be individually disabled by writing to 1 the appropriate bit in the RX_INTMASK_CLEAR register.
- The receive threshold pulse interrupt could be disabled by clearing to Zero the corresponding bits of the RX_THRESH_EN. The software could still poll for the RX_INTSTAT_RAW and INTSTAT_MASKED registers if the corresponding interrupts are enabled.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.3.5.4 Miscellaneous Pulse Interrupt (MISC_PULSE)

The MISC_PULSE interrupt is an immediate pulse interrupt selected from the miscellaneous interrupts (**EVNT_PEND**,STAT_PEND, HOST_PEND, MDIO_LINKINT, MDIO_USERINT). The miscellaneous interrupt(s) is selected by setting one or more bits in the **Misc_En[4:0]** register. The masked interrupt status can be read in the **Misc_Stat[4:0]** address location. Upon reception of an interrupt, software should perform the following:

- Read the **Misc_Stat[4:0]** address location to determine which caused the interrupt.
- Process the interrupt.
- Write the appropriate value (0x3) to the CPDMA_**EOI_VECTOR**
- **MDIO_LINTINT[1]** and **MDIO_USERINT[1]** are unused.

13.3.5.4.1 EVNT_PEND (CPTS_PEND) Interrupt

See CPTS TIME SYNC for more details on this interrupt.

13.3.5.4.2 Statistics Interrupt

The statistics level interrupt (**STAT_PEND)** will be asserted if enabled when any statistics value is greater than or equal to 0x80000000. The statistics interrupt is cleared by writing to decrement all statistics values greater than 0x80000000 (such that their new values are less than 0x80000000). The raw and masked statistics interrupt status may be read by reading the TX_**IntStat_Raw** and TX_**IntStat_Masked** registers respectively

13.3.5.4.3 Host Error Interrupt

The host error interrupt (HOST_PEND) will be asserted if enabled when a host error is detected during transmit or receive CPDMA transactions. The host error interrupt is intended for software debug, and is cleared by a warm reset or a system reset. The raw and masked statistics interrupt status can be read by reading the TX_INTSTAT_RAW and TXINTSTAT_MASKED registers, respectively.

The following list shows the transmit host error conditions:

- SOP error
- OWNERSHIP bit not set in SOP buffer
- next buffer descriptor pointer without EOP set to 0
- buffer pointer set to 0
- buffer length set to 0
- packet length error

The receive host error conditions are show in the following list:

- Ownership bit not set in input buffer.
- Zero buffer pointer.
- Zero buffer Length on non-SOP descriptor.
- SOP buffer length not greater than offset.

The host error interrupt is disabled by clearing to 0 the appropriate bit in the CPDMA_INTMASK_CLEAR register.

13.3.5.4.4 MDIO Interrupts

MDIO_LINKINT is set if there is a change in the link state of the PHY corresponding to the address in the PHYADDRMON field of the MDIOUSERPHYSEL*n* register and the corresponding LINKINTENB bit is set.The MDIO_LINKINT event is also captured in the MDIOLINKINTMASKED register.When the GO bit in the MDIOUSERREGISTER*n* registers transitions from 1 to 0, indicating the completion of a user access, and the corresponding USERINTMASKSET bit in the

MDIOUSERINTMASKSET register is set, the MDIO_USERINT signal is asserted 1. The MDIO_USERINT event is also captured in the MDIOUSERINTMASKED register.

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

To enable the miscellaneous pulse interrupt:

The miscellaneous interrupt(s) is selected by setting one or more bits in the miscellaneous interrupt enable register (MISC_EN).

- The Statistics interrupt is enabled by setting to 1 the STAT_INT_MASK bit in the CPDMAINTMASK_SET register.
- The HOST_PEND is enabled by setting to 1 the HOST_ERR_INTMASK in the CPDMAINTMASK_SET register.

Upon receiving of an interrupt, software should perform the following:

- Read the MISC_STAT bit address location to determine the source of the interrupt.
- Process the interrupt.
- Write the value 3h to the CPDMA_EOI_VECTOR register.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.3.6 Interrupt Pacing

The receive and transmit pulse interrupts can be paced. The receive threshold and miscellaneous interrupts are not paced. The Interrupt pacing feature limits the number of interrupts that occur during a given period of time. For heavily loaded systems in which interrupts can occur at a very high rate (e.g. 148,800 packets per second for Ethernet), the performance benefit is significant due to minimizing the overhead associated with servicing each interrupt. Interrupt pacing increases the CPU cache hit ratio by minimizing the number of times that large interrupt service routines are moved to and from the CPU instruction cache.

Each CPU receive and transmit pulse interrupt contains an interrupt pacing sub-block (six total). Each sub-block is disabled by default allowing the selected interrupt inputs to pass through unaffected. The interrupt pacing module counts the number of interrupts that occur over a 1ms interval of time. At the end of each 1ms interval, the current number of interrupts is compared with a target number of interrupts (specified by the associated maximum number of interrupts register).

Based on the results of the comparison, the length of time during which interrupts are blocked is dynamically adjusted. The 1ms interval is derived from a 4us pulse that is created from a prescale counter whose value is set in the **int_prescale** value in the **Int_Control** register. The **int_prescale** value should be written with the number of **VBUSP_CLK** periods in 4us. The pacing timer determines the interval during which interrupts are blocked and decrements every 4us. It is reloaded each time a zero count is reached. The value loaded into the pacing timer is calculated by hardware every 1ms according to the following algorithm:

```
if (intr_count > 2*intr_max)
pace_timer = 255;
else if (intr_count > 1.5*intr_max)
pace_timer = last_pace_timer*2 + 1;
else if (intr_count > 1.0*intr_max)
pace_timer = last_pace_timer + 1;
else if (intr_count > 0.5*intr_max)
pace_timer = last_pace_timer - 1;
else if (intr_count != 0)
pace_timer = last_pace_timer/2;
else
pace_timer = 0;
```

If the rate of interrupt inputs is much less than the target interrupt rate specified in the associated maximum interrupts register, then the interrupt is not blocked. If the interrupt rate is greater than the target rate, the interrupt will be “paced” at the rate specified in the interrupt maximum register. The interrupt maximum register should be written with a value between 2 and 63 inclusive indicating the target number of interrupts per milli-second.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.3.7 Reset Isolation

Reset isolation for the Ethernet switch on Device is that the switch function of the ethernet IP remains active even in case of all device resets except for POR pin reset and ICEPICK COLD reset. Packet traffic to/from the 3PSW host will be flushed/dropped, but the ethernet switch will remain operational for all traffic between external devices on the switch even though the device is under-going a device reset.Pin mux configuration for ethernet related IO and reference clocks needed by the Ethernet switch IP to be active is controlled by a protected control module bit. If reset isolation is enabled, then only a POR pin or ICEPICK COLD reset event should fully reset the Ethernet switch IP including the actual switch and also the reference clocks and pin mux control specifically associated with the Ethernet IP.

13.3.7.1 Modes of Operation

The device has two modes of operation concerning the reset of the 3PSW Ethernet switch.

The mode is controlled by the ISO_CONTROL bit in **RESET_ISO** register of the **Device control module**.This bit should default to '0'. Any modification of this bit first requires writing an unlock pattern to lock register in Device control module. After modification, the bit should again be locked by writing appropriate value to the lock register (see Device control module for details). Writes to the ISO_CONTROL bit and to the lock register must all be supervisor mode writes.

Mode1: ISO_CONTROL=0 (reset isolation disabled)

- This mode is selected when ISO_CONTROL bit of control module is = 0. This should be the default state of the bit after control module reset.
- Upon any device level resets, the entire CPSW_3GSS_R IP, SATA_SS, SATA SERDES pll, L3/L4, control module (including all pin mux control and the ISO_CONTROL bit) is immediately reset.

Mode2: ISO_CONTROL=1 (reset isolation enabled)

- This mode is selected when ISO_CONTROL bit of control module is = 1.
- Upon any device reset source other than POR pin or ICEPICK cold (so this includes SW global cold, any watchdog reset, warm RESETn pin, ICEPICK warm, SW global warm or security violation), the following should be true:
  - The CPSW_3GSS_R is put into ‘isolate’ mode and non-switch related portions of the IP are reset.
  - The 50-MHz and 125-MHz reference clocks to the 3PSW Ethernet Subsystem remains active throughout the entire reset condition.
  - The control for pin multiplexing for all of the signals should maintain their current configuration throughout the entire reset condition.
  - The reset isolated logic inside 3PSW Ethernet Subsystem IP which maintains the switch functionality
  -
- Upon any cold reset sources, the entire 3PSW Ethernet Subsystem, SATA SERDES pll, control module (including all pin mux control and the ISO_CONTROL bit itself) is reset.

For more details on the register configuration, see the Device Control Module reset_iso register.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.3.8 CPSW_3G Network Statistics

The CPSW_3G has a set of statistics that record events associated with frame traffic on selected switch ports. The statistics values are cleared to zero 38 clocks after the rising edge of **VBUSP_RST_N**. When one or more port enable bits **(stat_port_en[2:0])** are set, all statistics registers are write to decrement. The value written will be subtracted from the register value with the result being stored in the register. If a value greater than the statistics value is written, then zero will be written to the register (writing 0xffffffff will clear a statistics location). When all port enable bits are cleared to zero, all statistics registers are read/write (normal write direct, so writing 0x00000000 will clear a statistics location). All write accesses must be 32-bit accesses. In the below statistics descriptions, “the port” refers to any enabled port (with a corresponding set **stat_port_en[2:0]** bit).

The statistics interrupt (**STAT_PEND)** will be issued if enabled when any statistics value is greater than or equal to 0x80000000. The statistics interrupt is removed by writing to decrement any statistics value greater than 0x80000000. The statistics are mapped into internal memory space and are 32-bits wide. All statistics rollover from 0xFFFFFFFF to 0x00000000.

See the Section CPSW Memory Map for more details on the register description.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.4 Software Operation

13.4.1 Transmit Operation

After reset the host must write zeroes to all Tx DMA State head descriptor pointers. The Tx port may then be enabled. To initiate packet transmission the host constructs transmit queues in memory (one or more packets for transmission) and then writes the appropriate Tx DMA state head descriptor pointers. For each buffer added to a transmit queue, the host must initialize the Tx buffer descriptor values as follows:

1. Write the Next Descriptor Pointer with the 32-bit aligned address of the next descriptor in the queue (zero if last descriptor).
2. Write the Buffer Pointer with the byte aligned address of the buffer data.
3. Write the Buffer Length with the number of bytes in the buffer.
4. Write the Buffer Offset with the number of bytes in the offset to the data (nonzero with SOP only).
5. Set the SOP, EOP, and Ownership bits as appropriate.
6. Clear the End Of Queue bit.

The port begins Tx packet transmission on a given channel when the host writes the channel's Tx queue head descriptor pointer with the address of the first buffer descriptor in the queue (nonzero value). Each channel may have one or more queues, so each channel may have one or more head descriptor pointers. The first buffer descriptor for each Tx packet must have the Start of Packet (SOP) bit and the Ownership bit set to one by the host. The last buffer descriptor for each Tx packet must have the End of Packet (EOP) bit set to one by the host.

The port will transmit packets until all queued packets have been transmitted and the queue(s) are empty. When each packet transmission is complete, the port will clear the Ownership bit in the packet's SOP buffer descriptor and issue an interrupt to the host by writing the packet's last buffer descriptor address to the queue's Tx DMA State Completion Pointer. The interrupt is generated by the write, regardless of the value written.

When the last packet in a queue has been transmitted, the port sets the End Of Queue bit in the EOP buffer descriptor, clears the Ownership bit in the SOP Descriptor, zeroes the appropriate DMA state head descriptor pointer, and then issues a Tx interrupt to the host by writing to the queue's associated Tx completion pointer (address of the last buffer descriptor processed by the port). The port issues a maskable level interrupt (which may then be routed through external interrupt control logic to the host).

On interrupt from the port, the host processes the buffer queue, detecting transmitted packets by the status of the Ownership bit in the SOP buffer descriptor. If the Ownership bit is cleared to zero, then the packet has been transmitted and the host may reclaim the buffers associated with the packet. The host continues queue processing until the end of the queue or until a SOP buffer descriptor is read that contains a set Ownership bit indicating that the packet transmission is not complete.

The host determines that all packets in the queue have been transmitted when the last packet in the queue has a cleared Ownership bit in the SOP buffer descriptor, the End of Queue bit is set in the last packet EOP buffer descriptor, and the Next Descriptor Pointer of the last packet EOP buffer descriptor is zero. The host acknowledges an interrupt by writing the address of the last buffer descriptor to the queue's associated Tx Completion Pointer in the Tx DMA State.

If the host written buffer address value is different from the buffer address written by the port, then the level interrupt remains asserted. If the host written buffer address value is equal to the port written value, then the level interrupt is deasserted. The port write to the completion pointer actually stores the value in the state register (ram). The host written value is actually not written to the register location. The host written value is compared to the register contents (which was written by the port) and if the two values are equal, the interrupt is removed, otherwise the interrupt remains asserted. The host may process multiple packets previous to acknowledging an interrupt, or the host may acknowledge interrupts for every packet.

A misqueued packet condition may occur when the host adds a packet to a queue for transmission as the port finishes transmitting the previous last packet in the queue. The misqueued packet is detected by the host when queue processing detects a cleared Ownership bit in the SOP buffer descriptor, a set End of Queue bit in the EOP buffer descriptor, and a nonzero Next Descriptor Pointer in the EOP buffer descriptor.

Copyright © 2011, Texas Instruments Incorporated

A misqueued packet means that the port read the last EOP buffer descriptor before the host added the new last packet to the queue, so the port determined queue empty just before the last packet was added. The host corrects the misqueued packet condition by initiating a new packet transfer for the misqueued packet by writing the misqueued packet's SOP buffer descriptor address to the appropriate DMA State Tx Queue head Descriptor Pointer.

The host may add packets to the tail end of an active Tx queue at any time by writing the Next Descriptor Pointer to the current last descriptor in the queue. If a Tx queue is empty (inactive), the host may initiate packet transmission at any time by writing the appropriate Tx DMA State head descriptor pointer.

The host software should always check for and reinitiate transmission for misqueued packets during queue processing on interrupt from the port. In order to preclude software underrun, the host should avoid adding buffers to an active queue for any Tx packet that is not complete and ready for transmission.

The port determines that a packet is the last packet in the queue by detecting the End of Packet bit set with a zero Next Descriptor Pointer in the packet buffer descriptor. If the End of Packet bit is set and the Next Descriptor Pointer is nonzero, then the queue still contains one or more packets to be transmitted. If the EOP bit is set with a zero Next Descriptor Pointer, then the port will set the EOQ bit in the packet's EOP buffer descriptor and then zero the appropriate head descriptor pointer previous to interrupting the port (by writing the completion pointer) when the packet transmission is complete.

Figure 13-13.

13.4.2 Receive Operation

Figure 13-14.

After reset the host must write zeroes to all Rx DMA State head descriptor pointers. The Rx port may then be enabled. To initiate packet reception, the host constructs receive queues in memory and then writes the appropriate Rx DMA state head descriptor pointer. For each Rx buffer descriptor added to the queue, the host must initialize the Rx buffer descriptor values as follows:

- Write the Next Descriptor Pointer with the 32-bit aligned address of the next descriptor in the queue (zero if last descriptor)
- Write the Buffer Pointer with the byte aligned address of the buffer data
- Clear the Offset field
- Write the Buffer Length with the number of bytes in the buffer
- Clear the SOP, EOP, and EOQ bits
- Set the Ownership bit

The host enables packet reception on a given channel by writing the address of the first buffer descriptor in the queue (nonzero value) to the channel's head descriptor pointer in the channel's Rx DMA state. When packet reception begins on a given channel, the port fills each Rx buffer with data in order starting with the first buffer and proceeding through the Rx queue. If the Buffer Offset in the Rx DMA State is nonzero, then the port will begin writing data after the offset number of bytes in the SOP buffer. The port performs the following operations at the end of each packet reception:

- Overwrite the buffer length in the packet's EOP buffer descriptor with the number of bytes actually received in the packet's last buffer. The host initialized value is the buffer size. The overwritten value will be less than or equal to the host initialized value.
- Set the EOP bit in the packet's EOP buffer descriptor.
- Set the EOQ bit in the packet's EOP buffer descriptor if the current packet is the last packet in the queue.
- Overwrite the packet's SOP buffer descriptor Buffer Offset with the Rx DMA state value (the host initialized the buffer descriptor Buffer Offset value to zero). All non SOP buffer descriptors must have a zero Buffer Offset initialized by the host.
- Overwrite the packet's SOP buffer descriptor buffer length with the number of valid data bytes in the buffer. If the buffer is filled up, the buffer length will be the buffer size minus buffer offset.

Copyright © 2011, Texas Instruments Incorporated

- Set the SOP bit in the packet's SOP buffer descriptor.
- Write the SOP buffer descriptor Packet Length field.
- Clear the Ownership bit in the packet's SOP buffer descriptor.
- Issue an Rx host interrupt by writing the address of the packet's last buffer descriptor to the queue's Rx DMA State Completion Pointer. The interrupt is generated by the write to the Rx DMA State Completion Pointer address location, regardless of the value written.

On interrupt the host processes the Rx buffer queue detecting received packets by the status of the Ownership bit in each packet's SOP buffer descriptor. If the Ownership bit is cleared then the packet has been completely received and is available to be processed by the host.

The host may continue Rx queue processing until the end of the queue or until a buffer descriptor is read that contains a set Ownership bit indicating that the next packet's reception is not complete. The host determines that the Rx queue is empty when the last packet in the queue has a cleared Ownership bit in the SOP buffer descriptor, a set End of Queue bit in the EOP buffer descriptor, and the Next Descriptor Pointer in the EOP buffer descriptor is zero.

A misqueued buffer may occur when the host adds buffers to a queue as the port finishes the reception of the previous last packet in the queue. The misqueued buffer is detected by the host when queue processing detects a cleared Ownership bit in the SOP buffer descriptor, a set End of Queue bit in the EOP buffer descriptor, and a nonzero Next Descriptor Pointer in the EOP buffer descriptor.

A misqueued buffer means that the port read the last EOP buffer descriptor before the host added buffer descriptor(s) to the queue, so the port determined queue empty just before the host added more buffer descriptor(s). In the transmit case, the packet transmission is delayed by the time required for the host to determine the condition and reinitiate the transaction, but the packet is not actually lost. In the receive case, receive overrun condition may occur in the misqueued buffer case.

If a new packet reception is begun during the time that the port has determined the end of queue condition, then the received packet will overrun (start of packet overrun). If the misqueued buffer occurs during the middle of a packet reception then middle of packet overrun may occur. If the misqueued buffer occurs after the last packet has completed, and is corrected before the next packet reception begins, then overrun will not occur. The host acts on the misqueued buffer condition by writing the added buffer descriptor address to the appropriate Rx DMA State Head Descriptor Pointer.

13.4.3 Initializing the MDIO Module

The following steps are performed by the application software or device driver to initialize the MDIO device:

1. 1. Configure the PREAMBLE and CLKDIV bits in the MDIO control register (MDIOCONTROL).
2. Enable the MDIO module by setting the ENABLE bit in MDIOCONTROL.
3. The MDIO PHY alive status register (MDIOALIVE) can be read in polling fashion until a PHY connected to the system responded, and the MDIO PHY link status register (MDIOLINK) can determine whether this PHY already has a link.
4. Setup the appropriate PHY addresses in the MDIO user PHY select register (MDIOUSERPHYSEL*n*), and set the LINKINTENB bit to enable a link change event interrupt if desirable.

- If an interrupt on general MDIO register access is desired, set the corresponding bit in the MDIO user command complete interrupt mask set register (MDIOUSERINTMASKSET) to use the MDIO user access register (MDIOUSERACCESS*n*). Since only one PHY is used in this device, the application software can use one MDIOUSERACCESS*n* to trigger a completion interrupt; the other MDIOUSERACCESS*n* is not setup.

13.4.4 Writing Data to a PHY Register

The MDIO module includes a user access register (MDIOUSERACCESS*n*) to directly access a specified PHY device.To write a PHY register, perform the following:

1. Check to ensure that the GO bit in the MDIO user access register (MDIOUSERACCESS*n*) is cleared.

2. Write to the GO, WRITE, REGADR, PHYADR, and DATA bits in MDIOUSERACCESS*n* corresponding to the PHY and PHY register you want to write.

Copyright © 2011, Texas Instruments Incorporated

3. The write operation to the PHY is scheduled and completed by the MDIO module. Completion of the write operation can be determined by polling the GO bit in MDIOUSERACCESS*n* for a 0.

4. Completion of the operation sets the corresponding USERINTRAW bit (0 or 1) in the MDIO user command complete interrupt register (MDIOUSERINTRAW) corresponding to USERACCESS*n* used. If interrupts have been enabled on this bit using the MDIO user command complete interrupt mask set register (MDIOUSERINTMASKSET), then the bit is also set in the MDIO user command complete interrupt register (MDIOUSERINTMASKED) and an interrupt is triggered on the CPU.

13.4.5 Reading Data from a PHY Register

The MDIO module includes a user access register (MDIOUSERACCESS*n*) to directly access a specified PHY device. To read a PHY register, perform the following:

1. Check to ensure that the GO bit in the MDIO user access register (MDIOUSERACCESS*n*) is cleared.

2. Write to the GO, REGADR, and PHYADR bits in MDIOUSERACCESS*n* corresponding to the PHY and PHY register you want to read.

3. The read data value is available in the DATA bits in MDIOUSERACCESS*n* after the module completes the read operation on the serial bus. Completion of the read operation can be determined by polling the GO and ACK bits in MDIOUSERACCESS*n*. After the GO bit has cleared, the ACK bit is set on a successful read.

4. Completion of the operation sets the corresponding USERINTRAW bit (0 or 1) in the MDIO user command complete interrupt register (MDIOUSERINTRAW) corresponding to USERACCESS*n* used. If interrupts have been enabled on this bit using the MDIO user command complete interrupt mask set register (MDIOUSERINTMASKSET), then the bit is also set in the MDIO user command complete interrupt register (MDIOUSERINTMASKED) and an interrupt is triggered on the CPU.

13.4.6 Initialization and Configuration of CPSW

To configure the 3PSW Ethernet Subsystem for operation the host must perform the following:

- Select the Interface (GMII/RGMII/MII) Mode. (Refer to section Error: Reference source not found for details)
- Configure pads (PIN muxing) as per the Interface Selected.(Refer to Device Data Sheet Document for Pin Mux Details)
- Enable the 3PSW Ethernet Subsystem Clocks. (Refer to section Error: Reference source not found for details)
- Configure the PRCM Registers CM_ETHERNET_CLKSTCTRL, CM_ALWON_ETHERNET_0_CLKCTRL to enable power and clocks to 3PSW Ethernet Subsystem. (Refer to **PRCM** Document for register details)
- Apply Soft Reset to 3psw Subsytem,cpsw_3g, cpgmac_sl1/2 and cpdma(Refer to Soft Reset Registers in Section 4,5,6)
- Initialize the HDPs (Header Description Pointer) and CPs (Completion Pointer) to NULL
- Configure the Interrupts (Refer to section.**Interrupts**)
- Configure the CPSW_3G Control register.
- Configure the Statistics Port Enable register
- Configure the ALE. (Refer to section)
- Configure the MDIO (Refer to section **MDIO Software Interface**)
- Configure the CPDMA receive DMA controller(Refer to section Error: Reference source not found)
- Configure the CPDMA transmit DMA controller.(Refer to section Error: Reference source not found)
- Configure the CPPI Tx and Rx Descriptors(Refer to section Transmit Operation and Receive Operation)
- Configure CPGMAC_SL1 and CPGMAC_SL2 as per the desired mode of operations.
- Start up RX and TX DMA (Write to HDP of Rx and Tx).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

- Wait for the completion of Transfer(HDP cleared to Zero)

13.5 CPSW_ALE Registers

Table 13-21 lists the memory-mapped registers for the CPSW_ALE. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 13-21 should be considered as reserved locations and the register contents should not be modified.

Table 13-21. CPSW_ALE REGISTERS

Offset	Acronym	Register Description	Section
0h	IDVER	ADDRESS LOOKUP ENGINE ID/VERSION REGISTER	Section 13.11.1
10h	PRESCALE	ADDRESS LOOKUP ENGINE PRESCALE REGISTER	Section 13.5.2
18h	UNKNOWN_VLAN	ADDRESS LOOKUP ENGINE UNKNOWN VLAN REGISTER	Section 13.5.3
20h	TBLCTL	ADDRESS LOOKUP ENGINE TABLE CONTROL	Section 13.5.4
34h	TBLW2	ADDRESS LOOKUP ENGINE TABLE WORD 2 REGISTER	Section 13.5.5
38h	TBLW1	ADDRESS LOOKUP ENGINE TABLE WORD 1 REGISTER	Section 13.5.6
3Ch	TBLW0	ADDRESS LOOKUP ENGINE TABLE WORD 0 REGISTER	Section 13.5.7
40h	PORTCTL0	ADDRESS LOOKUP ENGINE PORT 0 CONTROL REGISTER	Section 13.5.8
44h	PORTCTL1	ADDRESS LOOKUP ENGINE PORT 1 CONTROL REGISTER	Section 13.5.9
48h	PORTCTL2	ADDRESS LOOKUP ENGINE PORT 2 CONTROL REGISTER	Section 13.5.10
4Ch	PORTCTL3	ADDRESS LOOKUP ENGINE PORT 3 CONTROL REGISTER	Section 13.5.11
8h	CONTROL	ADDRESS LOOKUP ENGINE CONTROL REGISTER	Section 13.11.30
50h	PORTCTL4	ADDRESS LOOKUP ENGINE PORT 4 CONTROL REGISTER	Section 13.5.13
54h	PORTCTL5	ADDRESS LOOKUP ENGINE PORT 5 CONTROL REGISTER	Section 13.5.14

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.5.1 IDVER Register (offset = 0h) [reset = 290104h]

ADDRESS LOOKUP ENGINE ID/VERSION REGISTER

IDVER is shown in Figure 13-198 and described in Table 13-211.

Figure 13-15. IDVER Register

31	30	29	28	27	26	25	24
IDENT							
R-29h							

23	22	21	20	19	18	17	16
IDENT							
R-29h							

15	14	13	12	11	10	9	8
MAJ_VER							
R-1h							

7	6	5	4	3	2	1	0
MINOR_VER							
R-4h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-22. IDVER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	IDENT	R	29h	ALE Identification Value
15-8	MAJ_VER	R	1h	ALE Major Version Value
7-0	MINOR_VER	R	4h	ALE Minor Version Value

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.5.2 PRESCALE Register (offset = 10h) [reset = 0h]

ADDRESS LOOKUP ENGINE PRESCALE REGISTER

PRESCALE is shown in Figure 13-16 and described in Table 13-23.

Figure 13-16. PRESCALE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												PRESCALE																			
																					R/W-0h										

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-23. PRESCALE Register Field Descriptions

Bit	Field	Type	Reset	Description
19-0	PRESCALE	R/W	0h	ALE Prescale Register - The input clock is divided by this value for use in the multicast/broadcast rate limiters. The minimum operating value is 0x10. The prescaler is off when the value is zero.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.5.3 UNKNOWN_VLAN Register (offset = 18h) [reset = 0h]

ADDRESS LOOKUP ENGINE UNKNOWN VLAN REGISTER

UNKNOWN_VLAN is shown in Figure 13-17 and described in Table 13-24.

Figure 13-17. UNKNOWN_VLAN Register

31	30	29	28	27	26	25	24
Reserved		UNKNOWN_FORCE_UNTAGGED_EGRESS					
				R/W-0h			

23	22	21	20	19	18	17	16
Reserved		UNKNOWN_REG_MCAST_FLOOD_MASK					
				R/W-0h			

15	14	13	12	11	10	9	8
Reserved		UNKNOWN_MCAST_FLOOD_MASK					
				R/W-0h			

7	6	5	4	3	2	1	0
Reserved		UNKNOWN_VLAN_MEMBER_LIST					
				R/W-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-24. UNKNOWN_VLAN Register Field Descriptions

Bit	Field	Type	Reset	Description
29-24	UNKNOWN_FORCE_UNTAGGED_EGRESS	R/W	0h	Unknown VLAN Force Untagged Egress.
21-16	UNKNOWN_REG_MCAST_FLOOD_MASK	R/W	0h	Unknown VLAN Registered Multicast Flood Mask
13-8	UNKNOWN_MCAST_FLOOD_MASK	R/W	0h	Unknown VLAN Multicast Flood Mask
5-0	UNKNOWN_VLAN_MEMBER_LIST	R/W	0h	Unknown VLAN Member List

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.5.4 TBLCTL Register (offset = 20h) [reset = 0h]

ADDRESS LOOKUP ENGINE TABLE CONTROL

TBLCTL is shown in Figure 13-18 and described in Table 13-25.

Figure 13-18. TBLCTL Register

31	30	29	28	27	26	25	24
WRITE_RDZ	Reserved						
R/W-0h							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved						ENTRY_POINTER	
						R/W-0h	

7	6	5	4	3	2	1	0
ENTRY_POINTER							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-25. TBLCTL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	WRITE_RDZ	R/W	0h	Write Bit - This bit is always read as zero. Writing a 1 to this bit causes the three table word register values to be written to the entry_pointer location in the address table. Writing a 0 to this bit causes the three table word register values to be loaded from the entry_pointer location in the address table so that they may be subsequently read. A read of any ALE address location will be stalled until the read or write has completed.
9-0	ENTRY_POINTER	R/W	0h	Table Entry Pointer - The entry_pointer contains the table entry value that will be read/written with accesses to the table word registers.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.5.5 TBLW2 Register (offset = 34h) [reset = 0h]

ADDRESS LOOKUP ENGINE TABLE WORD 2 REGISTER

TBLW2 is shown in Figure 13-19 and described in Table 13-26.

Figure 13-19. TBLW2 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								ENTRY71-64							
																								R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-26. TBLW2 Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	ENTRY71-64	R/W	0h	Table entry bits 71:64

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.5.6 TBLW1 Register (offset = 38h) [reset = 0h]

ADDRESS LOOKUP ENGINE TABLE WORD 1 REGISTER

TBLW1 is shown in Figure 13-20 and described in Table 13-27.

Figure 13-20. TBLW1 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENTRY63_32																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-27. TBLW1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ENTRY63_32	R/W	0h	Table entry bits 63:32

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.5.7 TBLW0 Register (offset = 3Ch) [reset = 0h]

ADDRESS LOOKUP ENGINE TABLE WORD 0 REGISTER

TBLW0 is shown in Figure 13-21 and described in Table 13-28.

Figure 13-21. TBLW0 Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENTRY31_0																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-28. TBLW0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	ENTRY31_0	R/W	0h	Table entry bits 31:0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.5.8 PORTCTL0 Register (offset = 40h) [reset = 0h]

ADDRESS LOOKUP ENGINE PORT 0 CONTROL REGISTER

PORTCTL0 is shown in Figure 13-22 and described in Table 13-29.

Figure 13-22. PORTCTL0 Register

31	30	29	28	27	26	25	24
BCAST_LIMIT							
R/W-0h							

23	22	21	20	19	18	17	16
MCAST_LIMIT							
R/W-0h							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved		NO_SA_UPDATE	NO_LEARN	VID_INGRESS_CHECK	DROP_UNTAGGED	PORT_STATE	
		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-29. PORTCTL0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	BCAST_LIMIT	R/W	0h	Broadcast Packet Rate Limit - Each prescale pulse loads this field into the port broadcast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Broadcast rate limiting is enabled by a non-zero value in this field
23-16	MCAST_LIMIT	R/W	0h	Multicast Packet Rate Limit - Each prescale pulse loads this field into the port multicast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Multicast rate limiting is enabled by a non-zero value in this field.
5	NO_SA_UPDATE	R/W	0h	No Souce Address Update - When set the port is disabled from updating the source port number in an ALE table entry.
4	NO_LEARN	R/W	0h	No Learn Mode - When set the port is disabled from learning an address.
3	VID_INGRESS_CHECK	R/W	0h	VLAN ID Ingress Check - If VLAN not found then drop the packet. Packets with an unknown (default) VLAN will be dropped.
2	DROP_UNTAGGED	R/W	0h	Drop Untagged Packets - Drop non-VLAN tagged ingress packets.
1-0	PORT_STATE	R/W	0h	Port State 0 - Disabled 1 - Blocked 2 - Learn 3 - Forward

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.5.9 PORTCTL1 Register (offset = 44h) [reset = 0h]

ADDRESS LOOKUP ENGINE PORT 1 CONTROL REGISTER

PORTCTL1 is shown in Figure 13-23 and described in Table 13-30.

Figure 13-23. PORTCTL1 Register

31	30	29	28	27	26	25	24
BCAST_LIMIT							
R/W-0h							

23	22	21	20	19	18	17	16
MCAST_LIMIT							
R/W-0h							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved		NO_SA_UPDATE	NO_LEARN	VID_INGRESS_CHECK	DROP_UNTAGGED	PORT_STATE	
		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-30. PORTCTL1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	BCAST_LIMIT	R/W	0h	Broadcast Packet Rate Limit - Each prescale pulse loads this field into the port broadcast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Broadcast rate limiting is enabled by a non-zero value in this field
23-16	MCAST_LIMIT	R/W	0h	Multicast Packet Rate Limit - Each prescale pulse loads this field into the port multicast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Multicast rate limiting is enabled by a non-zero value in this field.
5	NO_SA_UPDATE	R/W	0h	No Souce Address Update - When set the port is disabled from updating the source port number in an ALE table entry.
4	NO_LEARN	R/W	0h	No Learn Mode - When set the port is disabled from learning an address.
3	VID_INGRESS_CHECK	R/W	0h	VLAN ID Ingress Check - If VLAN not found then drop the packet. Packets with an unknown (default) VLAN will be dropped.
2	DROP_UNTAGGED	R/W	0h	Drop Untagged Packets - Drop non-VLAN tagged ingress packets.
1-0	PORT_STATE	R/W	0h	Port State 0 - Disabled 1 - Blocked 2 - Learn 3 - Forward

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.5.10 PORTCTL2 Register (offset = 48h) [reset = 0h]

ADDRESS LOOKUP ENGINE PORT 2 CONTROL REGISTER

PORTCTL2 is shown in Figure 13-24 and described in Table 13-31.

Figure 13-24. PORTCTL2 Register

31	30	29	28	27	26	25	24
BCAST_LIMIT							
R/W-0h							

23	22	21	20	19	18	17	16
MCAST_LIMIT							
R/W-0h							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved		NO_SA_UPDATE	NO_LEARN	VID_INGRESS_CHECK	DROP_UNTAGGED	PORT_STATE	
		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-31. PORTCTL2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	BCAST_LIMIT	R/W	0h	Broadcast Packet Rate Limit - Each prescale pulse loads this field into the port broadcast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Broadcast rate limiting is enabled by a non-zero value in this field
23-16	MCAST_LIMIT	R/W	0h	Multicast Packet Rate Limit - Each prescale pulse loads this field into the port multicast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Multicast rate limiting is enabled by a non-zero value in this field.
5	NO_SA_UPDATE	R/W	0h	No Souce Address Update - When set the port is disabled from updating the source port number in an ALE table entry.
4	NO_LEARN	R/W	0h	No Learn Mode - When set the port is disabled from learning an address.
3	VID_INGRESS_CHECK	R/W	0h	VLAN ID Ingress Check - If VLAN not found then drop the packet. Packets with an unknown (default) VLAN will be dropped.
2	DROP_UNTAGGED	R/W	0h	Drop Untagged Packets - Drop non-VLAN tagged ingress packets.
1-0	PORT_STATE	R/W	0h	Port State 0 - Disabled 1 - Blocked 2 - Learn 3 - Forward

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.5.11 PORTCTL3 Register (offset = 4Ch) [reset = 0h]

ADDRESS LOOKUP ENGINE PORT 3 CONTROL REGISTER

PORTCTL3 is shown in Figure 13-25 and described in Table 13-32.

Figure 13-25. PORTCTL3 Register

31	30	29	28	27	26	25	24
BCAST_LIMIT							
R/W-0h							

23	22	21	20	19	18	17	16
MCAST_LIMIT							
R/W-0h							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved		NO_SA_UPDATE	NO_LEARN	VID_INGRESS_CHECK	DROP_UNTAGGED	PORT_STATE	
		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-32. PORTCTL3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	BCAST_LIMIT	R/W	0h	Broadcast Packet Rate Limit - Each prescale pulse loads this field into the port broadcast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Broadcast rate limiting is enabled by a non-zero value in this field
23-16	MCAST_LIMIT	R/W	0h	Multicast Packet Rate Limit - Each prescale pulse loads this field into the port multicast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Multicast rate limiting is enabled by a non-zero value in this field.
5	NO_SA_UPDATE	R/W	0h	No Souce Address Update - When set the port is disabled from updating the source port number in an ALE table entry.
4	NO_LEARN	R/W	0h	No Learn Mode - When set the port is disabled from learning an address.
3	VID_INGRESS_CHECK	R/W	0h	VLAN ID Ingress Check - If VLAN not found then drop the packet. Packets with an unknown (default) VLAN will be dropped.
2	DROP_UNTAGGED	R/W	0h	Drop Untagged Packets - Drop non-VLAN tagged ingress packets.
1-0	PORT_STATE	R/W	0h	Port State 0 - Disabled 1 - Blocked 2 - Learn 3 - Forward

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.5.12 CONTROL Register (offset = 8h) [reset = 0h]

ADDRESS LOOKUP ENGINE CONTROL REGISTER

CONTROL is shown in Figure 13-227 and described in Table 13-240.

Figure 13-26. CONTROL Register

31	30	29	28	27	26	25	24
ENABLE_ALE	CLEAR_TABLE	AGE_OUT_NOW	Reserved				
R/W-0h	R/W-0h	R/W-0h					

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							EN_P0_UNI_FLOOD
							R/W-0h

7	6	5	4	3	2	1	0
LEARN_NO_VID	EN_VID0_MODE	ENABLE_OUI_DENY	BYPASS	RATE_LIMIT_TX	VLAN_AWARE	ENABLE_AUTH_MODE	ENABLE_RATE_LIMIT
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-33. CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	ENABLE_ALE	R/W	0h	Enable ALE - 0 - Drop all packets 1 - Enable ALE packet processing
30	CLEAR_TABLE	R/W	0h	Clear ALE address table - Setting this bit causes the ALE hardware to write all table bit values to zero. Software must perform a clear table operation as part of the ALE setup/configuration process. Setting this bit causes all ALE accesses to be held up for 64 clocks while the clear is performed. Access to all ALE registers will be blocked (wait states) until the 64 clocks have completed. This bit cannot be read as one because the read is blocked until the clear table is completed at which time this bit is cleared to zero.
29	AGE_OUT_NOW	R/W	0h	Age Out Address Table Now - Setting this bit causes the ALE hardware to remove (free up) any ageable table entry that does not have a set touch bit. This bit is cleared when the age out process has completed. This bit may be read. The age out process takes 4096 clocks best case (no ale packet processing during ageout) and 66550 clocks absolute worst case.
8	EN_P0_UNI_FLOOD	R/W	0h	Enable Port 0 (Host Port) unicast flood 0 - do not flood unknown unicast packets to host port (p0) 1 - flood unknown unicast packets to host port (p0)
7	LEARN_NO_VID	R/W	0h	Learn No VID - 0 - VID is learned with the source address 1 - VID is not learned with the source address (source address is not tied to VID).
6	EN_VID0_MODE	R/W	0h	Enable VLAN ID = 0 Mode 0 - Process the packet with VID = PORT_VLAN[11:0]. 1 - Process the packet with VID = 0.
5	ENABLE_OUI_DENY	R/W	0h	Enable OUI Deny Mode - When set this bit indicates that a packet with a non OUI table entry matching source address will be dropped to the host unless the destination address matches a multicast table entry with the super bit set.
4	BYPASS	R/W	0h	ALE Bypass - When set, all packets received on ports 0 and 1 are sent to the host (only to the host).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 13-33. CONTROL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
3	RATE_LIMIT_TX	R/W	0h	Rate Limit Transmit mode - 0 - Broadcast and multicast rate limit counters are received port based 1 - Broadcast and multicast rate limit counters are transmit port based.
2	VLAN_AWARE	R/W	0h	ALE VLAN Aware - Determines what is done if VLAN not found. 0 - Flood if VLAN not found 1 - Drop packet if VLAN not found
1	ENABLE_AUTH_MODE	R/W	0h	Enable MAC Authorization Mode - Mac authorization mode requires that all table entries be made by the host software. There are no learned address in authorization mode and the packet will be dropped if the source address is not found (and the destination address is not a multicast address with the super table entry bit set). 0 - The ALE is not in MAC authorization mode 1 - The ALE is in MAC authorization mode
0	ENABLE_RATE_LIMIT	R/W	0h	Enable Broadcast and Multicast Rate Limit 0 - Broadcast/Multicast rates not limited 1 - Broadcast/Multicast packet reception limited to the port control register rate limit fields.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.5.13 PORTCTL4 Register (offset = 50h) [reset = 0h]

ADDRESS LOOKUP ENGINE PORT 4 CONTROL REGISTER

PORTCTL4 is shown in Figure 13-27 and described in Table 13-34.

Figure 13-27. PORTCTL4 Register

31	30	29	28	27	26	25	24
BCAST_LIMIT							
R/W-0h							

23	22	21	20	19	18	17	16
MCAST_LIMIT							
R/W-0h							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved		NO_SA_UPDATE	NO_LEARN	VID_INGRESS_CHECK	DROP_UNTAGGED	PORT_STATE	
		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-34. PORTCTL4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	BCAST_LIMIT	R/W	0h	Broadcast Packet Rate Limit - Each prescale pulse loads this field into the port broadcast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Broadcast rate limiting is enabled by a non-zero value in this field
23-16	MCAST_LIMIT	R/W	0h	Multicast Packet Rate Limit - Each prescale pulse loads this field into the port multicast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Multicast rate limiting is enabled by a non-zero value in this field.
5	NO_SA_UPDATE	R/W	0h	No Souce Address Update - When set the port is disabled from updating the source port number in an ALE table entry.
4	NO_LEARN	R/W	0h	No Learn Mode - When set the port is disabled from learning an address.
3	VID_INGRESS_CHECK	R/W	0h	VLAN ID Ingress Check - If VLAN not found then drop the packet. Packets with an unknown (default) VLAN will be dropped.
2	DROP_UNTAGGED	R/W	0h	Drop Untagged Packets - Drop non-VLAN tagged ingress packets.
1-0	PORT_STATE	R/W	0h	Port State 0 - Disabled 1 - Blocked 2 - Learn 3 - Forward

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.5.14 PORTCTL5 Register (offset = 54h) [reset = 0h]

ADDRESS LOOKUP ENGINE PORT 5 CONTROL REGISTER

PORTCTL5 is shown in Figure 13-28 and described in Table 13-35.

Figure 13-28. PORTCTL5 Register

31	30	29	28	27	26	25	24
BCAST_LIMIT							
R/W-0h							

23	22	21	20	19	18	17	16
MCAST_LIMIT							
R/W-0h							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved		NO_SA_UPDATE	NO_LEARN	VID_INGRESS_CHECK	DROP_UNTAGGED	PORT_STATE	
		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-35. PORTCTL5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	BCAST_LIMIT	R/W	0h	Broadcast Packet Rate Limit - Each prescale pulse loads this field into the port broadcast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Broadcast rate limiting is enabled by a non-zero value in this field
23-16	MCAST_LIMIT	R/W	0h	Multicast Packet Rate Limit - Each prescale pulse loads this field into the port multicast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Multicast rate limiting is enabled by a non-zero value in this field.
5	NO_SA_UPDATE	R/W	0h	No Souce Address Update - When set the port is disabled from updating the source port number in an ALE table entry.
4	NO_LEARN	R/W	0h	No Learn Mode - When set the port is disabled from learning an address.
3	VID_INGRESS_CHECK	R/W	0h	VLAN ID Ingress Check - If VLAN not found then drop the packet. Packets with an unknown (default) VLAN will be dropped.
2	DROP_UNTAGGED	R/W	0h	Drop Untagged Packets - Drop non-VLAN tagged ingress packets.
1-0	PORT_STATE	R/W	0h	Port State 0 - Disabled 1 - Blocked 2 - Learn

13.6 CPSW_CPDMA Registers

Table 13-36 lists the memory-mapped registers for the CPSW_CPDMA. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 13-36 should be considered as reserved locations and the register contents should not be modified.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 13-36. CPSW_CPDMA REGISTERS

Offset	Acronym	Register Description	Section
0h	TX_IDVER	CPDMA_REGS TX IDENTIFICATION AND VERSION REGISTER	Section 13.6.1
80h	TX_INTSTAT_RAW	CPDMA_INT TX INTERRUPT STATUS REGISTER (RAW VALUE)	Section 13.6.2
84h	TX_INTSTAT_MASKED	CPDMA_INT TX INTERRUPT STATUS REGISTER (MASKED VALUE)	Section 13.6.3
88h	TX_INTMASK_SET	CPDMA_INT TX INTERRUPT MASK SET REGISTER	Section 13.6.4
8Ch	TX_INTMASK_CLEAR	CPDMA_INT TX INTERRUPT MASK CLEAR REGISTER	Section 13.6.5
90h	CPDMA_IN_VECTOR	CPDMA_INT INPUT VECTOR (READ ONLY)	Section 13.6.6
94h	CPDMA_EOI_VECTOR	CPDMA_INT END OF INTERRUPT VECTOR	Section 13.6.7
10h	RX_IDVER	CPDMA_REGS RX IDENTIFICATION AND VERSION REGISTER	Section 13.6.8
A0h	RX_INTSTAT_RAW	CPDMA_INT RX INTERRUPT STATUS REGISTER (RAW VALUE)	Section 13.6.9
A4h	RX_INTSTAT_MASKED	CPDMA_INT RX INTERRUPT STATUS REGISTER (MASKED VALUE)	Section 13.6.10
A8h	RX_INTMASK_SET	CPDMA_INT RX INTERRUPT MASK SET REGISTER	Section 13.6.11
ACh	RX_INTMASK_CLEAR	CPDMA_INT RX INTERRUPT MASK CLEAR REGISTER	Section 13.6.12
B0h	DMA_INTSTAT_RAW	CPDMA_INT DMA INTERRUPT STATUS REGISTER (RAW VALUE)	Section 13.6.13
B4h	DMA_INTSTAT_MASKED	CPDMA_INT DMA INTERRUPT STATUS REGISTER (MASKED VALUE)	Section 13.6.14
B8h	DMA_INTMASK_SET	CPDMA_INT DMA INTERRUPT MASK SET REGISTER	Section 13.6.15
BCh	DMA_INTMASK_CLEAR	CPDMA_INT DMA INTERRUPT MASK CLEAR REGISTER	Section 13.6.16
C0h	RX0_PENDTHRESH	CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 0	Section 13.6.17
C4h	RX1_PENDTHRESH	CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 1	Section 13.6.18
14h	RX_CONTROL	CPDMA_REGS RX CONTROL REGISTER	Section 13.6.19
C8h	RX2_PENDTHRESH	CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 2	Section 13.6.20
CCh	RX3_PENDTHRESH	CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 3	Section 13.6.21
D0h	RX4_PENDTHRESH	CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 4	Section 13.6.22
D4h	RX5_PENDTHRESH	CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 5	Section 13.6.23
D8h	RX6_PENDTHRESH	CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 6	Section 13.6.24
DCh	RX7_PENDTHRESH	CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 7	Section 13.6.25
E0h	RX0_FREEBUFFER	CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 0	Section 13.6.26
E4h	RX1_FREEBUFFER	CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 1	Section 13.6.27
E8h	RX2_FREEBUFFER	CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 2	Section 13.6.28
ECh	RX3_FREEBUFFER	CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 3	Section 13.6.29
18h	RX_TEARDOWN	CPDMA_REGS RX TEARDOWN REGISTER	Section 13.6.30
F0h	RX4_FREEBUFFER	CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 4	Section 13.6.31

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 13-36. CPSW_CPDMA REGISTERS (continued)

Offset	Acronym	Register Description	Section
F4h	RX5_FREEBUFFER	CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 5	Section 13.6.32
F8h	RX6_FREEBUFFER	CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 6	Section 13.6.33
FCh	RX7_FREEBUFFER	CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 7	Section 13.6.34
A00h	TX0_HDP	CPDMA_STATERAM TX CHANNEL 0 HEAD DESC POINTER *	Section 13.6.35
A04h	TX1_HDP	CPDMA_STATERAM TX CHANNEL 1 HEAD DESC POINTER *	Section 13.6.36
A08h	TX2_HDP	CPDMA_STATERAM TX CHANNEL 2 HEAD DESC POINTER *	Section 13.6.37
A0Ch	TX3_HDP	CPDMA_STATERAM TX CHANNEL 3 HEAD DESC POINTER *	Section 13.6.38
A10h	TX4_HDP	CPDMA_STATERAM TX CHANNEL 4 HEAD DESC POINTER *	Section 13.6.39
A14h	TX5_HDP	CPDMA_STATERAM TX CHANNEL 5 HEAD DESC POINTER *	Section 13.6.40
A18h	TX6_HDP	CPDMA_STATERAM TX CHANNEL 6 HEAD DESC POINTER *	Section 13.6.41
A1Ch	TX7_HDP	CPDMA_STATERAM TX CHANNEL 7 HEAD DESC POINTER *	Section 13.6.42
A20h	RX0_HDP	CPDMA_STATERAM RX 0 CHANNEL 0 HEAD DESC POINTER *	Section 13.6.43
A24h	RX1_HDP	CPDMA_STATERAM RX 1 CHANNEL 1 HEAD DESC POINTER *	Section 13.6.44
A28h	RX2_HDP	CPDMA_STATERAM RX 2 CHANNEL 2 HEAD DESC POINTER *	Section 13.6.45
A2Ch	RX3_HDP	CPDMA_STATERAM RX 3 CHANNEL 3 HEAD DESC POINTER *	Section 13.6.46
A30h	RX4_HDP	CPDMA_STATERAM RX 4 CHANNEL 4 HEAD DESC POINTER *	Section 13.6.47
A34h	RX5_HDP	CPDMA_STATERAM RX 5 CHANNEL 5 HEAD DESC POINTER *	Section 13.6.48
A38h	RX6_HDP	CPDMA_STATERAM RX 6 CHANNEL 6 HEAD DESC POINTER *	Section 13.6.49
A3Ch	RX7_HDP	CPDMA_STATERAM RX 7 CHANNEL 7 HEAD DESC POINTER *	Section 13.6.50
A40h	TX0_CP	CPDMA_STATERAM TX CHANNEL 0 COMPLETION POINTER REGISTER	Section 13.6.51
A44h	TX1_CP	CPDMA_STATERAM TX CHANNEL 1 COMPLETION POINTER REGISTER *	Section 13.6.52
A48h	TX2_CP	CPDMA_STATERAM TX CHANNEL 2 COMPLETION POINTER REGISTER *	Section 13.6.53
A4Ch	TX3_CP	CPDMA_STATERAM TX CHANNEL 3 COMPLETION POINTER REGISTER *	Section 13.6.54
A50h	TX4_CP	CPDMA_STATERAM TX CHANNEL 4 COMPLETION POINTER REGISTER *	Section 13.6.55
A54h	TX5_CP	CPDMA_STATERAM TX CHANNEL 5 COMPLETION POINTER REGISTER *	Section 13.6.56
A58h	TX6_CP	CPDMA_STATERAM TX CHANNEL 6 COMPLETION POINTER REGISTER *	Section 13.6.57
A5Ch	TX7_CP	CPDMA_STATERAM TX CHANNEL 7 COMPLETION POINTER REGISTER *	Section 13.6.58
A60h	RX0_CP	CPDMA_STATERAM RX CHANNEL 0 COMPLETION POINTER REGISTER *	Section 13.6.59

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 13-36. CPSW_CPDMA REGISTERS (continued)

Offset	Acronym	Register Description	Section
A64h	RX1_CP	CPDMA_STATERAM RX CHANNEL 1 COMPLETION POINTER REGISTER *	Section 13.6.60
A68h	RX2_CP	CPDMA_STATERAM RX CHANNEL 2 COMPLETION POINTER REGISTER *	Section 13.6.61
A6Ch	RX3_CP	CPDMA_STATERAM RX CHANNEL 3 COMPLETION POINTER REGISTER *	Section 13.6.62
A70h	RX4_CP	CPDMA_STATERAM RX CHANNEL 4 COMPLETION POINTER REGISTER *	Section 13.6.63
A74h	RX5_CP	CPDMA_STATERAM RX CHANNEL 5 COMPLETION POINTER REGISTER *	Section 13.6.64
A78h	RX6_CP	CPDMA_STATERAM RX CHANNEL 6 COMPLETION POINTER REGISTER *	Section 13.6.65
A7Ch	RX7_CP	CPDMA_STATERAM RX CHANNEL 7 COMPLETION POINTER REGISTER *	Section 13.6.66
1Ch	CPDMA_SOFT_RESET	CPDMA_REGS SOFT RESET REGISTER	Section 13.6.67
20h	DMACONTROL	CPDMA_REGS CPDMA CONTROL REGISTER	Section 13.6.68
24h	DMASTATUS	CPDMA_REGS CPDMA STATUS REGISTER	Section 13.6.69
4h	TX_CONTROL	CPDMA_REGS TX CONTROL REGISTER	Section 13.6.70
28h	RX_BUFFER_OFFSET	CPDMA_REGS RECEIVE BUFFER OFFSET	Section 13.6.71
2Ch	EMCONTROL	CPDMA_REGS EMULATION CONTROL	Section 13.9.7
30h	TX_PRI0_RATE	CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 0 RATE	Section 13.6.73
34h	TX_PRI1_RATE	CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 1 RATE	Section 13.6.74
38h	TX_PRI2_RATE	CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 2 RATE	Section 13.6.75
3Ch	TX_PRI3_RATE	CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 3 RATE	Section 13.6.76
40h	TX_PRI4_RATE	CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 4 RATE	Section 13.6.77
44h	TX_PRI5_RATE	CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 5 RATE	Section 13.6.78
48h	TX_PRI6_RATE	CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 6 RATE	Section 13.6.79
4Ch	TX_PRI7_RATE	CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 7 RATE	Section 13.6.80
8h	TX_TEARDOWN	CPDMA_REGS TX TEARDOWN REGISTER	Section 13.6.81

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.1 TX_IDVER Register (offset = 0h) [reset = 180108h]

CPDMA_REGS TX IDENTIFICATION AND VERSION REGISTER

TX_IDVER is shown in Figure 13-29 and described in Table 13-37.

Figure 13-29. TX_IDVER Register

31	30	29	28	27	26	25	24
TX_IDENT							
R-18h							

23	22	21	20	19	18	17	16
TX_IDENT							
R-18h							

15	14	13	12	11	10	9	8
TX_MAJOR_VER							
R-1h							

7	6	5	4	3	2	1	0
TX_MINOR_VER							
R-8h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-37. TX_IDVER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TX_IDENT	R	18h	TX Identification Value
15-8	TX_MAJOR_VER	R	1h	TX Major Version Value - The value read is the major version number
7-0	TX_MINOR_VER	R	8h	TX Minor Version Value - The value read is the minor version number

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.2 TX_INTSTAT_RAW Register (offset = 80h) [reset = 0h]

CPDMA_INT TX INTERRUPT STATUS REGISTER (RAW VALUE)

TX_INTSTAT_RAW is shown in Figure 13-30 and described in Table 13-38.

Figure 13-30. TX_INTSTAT_RAW Register

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
TX7_PEND	TX6_PEND	TX5_PEND	TX4_PEND	TX3_PEND	TX2_PEND	TX1_PEND	TX0_PEND
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-38. TX_INTSTAT_RAW Register Field Descriptions

Bit	Field	Type	Reset	Description
7	TX7_PEND	R	0h	TX7_PEND raw int read (before mask).
6	TX6_PEND	R	0h	TX6_PEND raw int read (before mask).
5	TX5_PEND	R	0h	TX5_PEND raw int read (before mask).
4	TX4_PEND	R	0h	TX4_PEND raw int read (before mask).
3	TX3_PEND	R	0h	TX3_PEND raw int read (before mask).
2	TX2_PEND	R	0h	TX2_PEND raw int read (before mask).
1	TX1_PEND	R	0h	TX1_PEND raw int read (before mask).
0	TX0_PEND	R	0h	TX0_PEND raw int read (before mask).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.3 TX_INTSTAT_MASKED Register (offset = 84h) [reset = 0h]

CPDMA_INT TX INTERRUPT STATUS REGISTER (MASKED VALUE)

TX_INTSTAT_MASKED is shown in Figure 13-31 and described in Table 13-39.

Figure 13-31. TX_INTSTAT_MASKED Register

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
TX7_PEND	TX6_PEND	TX5_PEND	TX4_PEND	TX3_PEND	TX2_PEND	TX1_PEND	TX0_PEND
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-39. TX_INTSTAT_MASKED Register Field Descriptions

Bit	Field	Type	Reset	Description
7	TX7_PEND	R	0h	TX7_PEND masked interrupt read.
6	TX6_PEND	R	0h	TX6_PEND masked interrupt read.
5	TX5_PEND	R	0h	TX5_PEND masked interrupt read.
4	TX4_PEND	R	0h	TX4_PEND masked interrupt read.
3	TX3_PEND	R	0h	TX3_PEND masked interrupt read.
2	TX2_PEND	R	0h	TX2_PEND masked interrupt read.
1	TX1_PEND	R	0h	TX1_PEND masked interrupt read.
0	TX0_PEND	R	0h	TX0_PEND masked interrupt read.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.4 TX_INTMASK_SET Register (offset = 88h) [reset = 0h]

CPDMA_INT TX INTERRUPT MASK SET REGISTER

TX_INTMASK_SET is shown in Figure 13-32 and described in Table 13-40.

Figure 13-32. TX_INTMASK_SET Register

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
TX7_MASK	TX6_MASK	TX5_MASK	TX4_MASK	TX3_MASK	TX2_MASK	TX1_MASK	TX0_MASK
W-0h	R-0h	W-0h	R-0h	W-0h	R-0h	W-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-40. TX_INTMASK_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
7	TX7_MASK	W	0h	TX Channel 7 Mask - Write one to enable interrupt.
6	TX6_MASK	R	0h	TX Channel 6 Mask - Write one to enable interrupt.
5	TX5_MASK	W	0h	TX Channel 5 Mask - Write one to enable interrupt.
4	TX4_MASK	R	0h	TX Channel 4 Mask - Write one to enable interrupt.
3	TX3_MASK	W	0h	TX Channel 3 Mask - Write one to enable interrupt.
2	TX2_MASK	R	0h	TX Channel 2 Mask - Write one to enable interrupt.
1	TX1_MASK	W	0h	TX Channel 1 Mask - Write one to enable interrupt.
0	TX0_MASK	R	0h	TX Channel 0 Mask - Write one to enable interrupt.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.5 TX_INTMASK_CLEAR Register (offset = 8Ch) [reset = 0h]

CPDMA_INT TX INTERRUPT MASK CLEAR REGISTER

TX_INTMASK_CLEAR is shown in Figure 13-33 and described in Table 13-41.

Figure 13-33. TX_INTMASK_CLEAR Register

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
TX7_MASK	TX6_MASK	TX5_MASK	TX4_MASK	TX3_MASK	TX2_MASK	TX1_MASK	TX0_MASK
W-0h	R-0h	W-0h	R-0h	W-0h	R-0h	W-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-41. TX_INTMASK_CLEAR Register Field Descriptions

Bit	Field	Type	Reset	Description
7	TX7_MASK	W	0h	TX Channel 7 Mask - Write one to disable interrupt.
6	TX6_MASK	R	0h	TX Channel 6 Mask - Write one to disable interrupt.
5	TX5_MASK	W	0h	TX Channel 5 Mask - Write one to disable interrupt.
4	TX4_MASK	R	0h	TX Channel 4 Mask - Write one to disable interrupt.
3	TX3_MASK	W	0h	TX Channel 3 Mask - Write one to disable interrupt.
2	TX2_MASK	R	0h	TX Channel 2 Mask - Write one to disable interrupt.
1	TX1_MASK	W	0h	TX Channel 1 Mask - Write one to disable interrupt.
0	TX0_MASK	R	0h	TX Channel 0 Mask - Write one to disable interrupt.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.6 CPDMA_IN_VECTOR Register (offset = 90h) [reset = 0h]

CPDMA_INT INPUT VECTOR (READ ONLY)

CPDMA_IN_VECTOR is shown in Figure 13-34 and described in Table 13-42.

Figure 13-34. CPDMA_IN_VECTOR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMA_IN_VECTOR																															
R-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-42. CPDMA_IN_VECTOR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	DMA_IN_VECTOR	R	0h	DMA Input Vector - The value of DMA_In_Vector is reset to zero, but will change to the IN_VECTOR bus value one clock after reset is deasserted. Thereafter, this value will change to a new IN_VECTOR value one clock after the IN_VECTOR value changes.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.7 CPDMA_EOI_VECTOR Register (offset = 94h) [reset = 0h]

CPDMA_INT END OF INTERRUPT VECTOR

CPDMA_EOI_VECTOR is shown in Figure 13-35 and described in Table 13-43.

Figure 13-35. CPDMA_EOI_VECTOR Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																											DMA_EOI_VECTOR				
																											R/W-0h				

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-43. CPDMA_EOI_VECTOR Register Field Descriptions

Bit	Field	Type	Reset	Description
4-0	DMA_EOI_VECTOR	R/W	0h	DMA End of Interrupt Vector - The EOI_VECTOR(4:0) pins reflect the value written to this location one CLK cycle after a write to this location. The EOI_WR signal is asserted for a single clock cycle after a latency of two CLK cycles when a write is performed to this location.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.8 RX_IDVER Register (offset = 10h) [reset = C0107h]

CPDMA_REGS RX IDENTIFICATION AND VERSION REGISTER

RX_IDVER is shown in Figure 13-36 and described in Table 13-44.

Figure 13-36. RX_IDVER Register

31	30	29	28	27	26	25	24
RX_IDENT							
R-Ch							

23	22	21	20	19	18	17	16
RX_IDENT							
R-Ch							

15	14	13	12	11	10	9	8
RX_MAJOR_VER							
R-1h							

7	6	5	4	3	2	1	0
RX_MINOR_VER							
R-7h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-44. RX_IDVER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	RX_IDENT	R	Ch	RX Identification Value
15-8	RX_MAJOR_VER	R	1h	RX Major Version Value
7-0	RX_MINOR_VER	R	7h	RX Minor Version Value

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.9 RX_INTSTAT_RAW Register (offset = A0h) [reset = 0h]

CPDMA_INT RX INTERRUPT STATUS REGISTER (RAW VALUE)

RX_INTSTAT_RAW is shown in Figure 13-37 and described in Table 13-45.

Figure 13-37. RX_INTSTAT_RAW Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
RX7_THRESH_PEND	RX6_THRESH_PEND	RX5_THRESH_PEND	RX4_THRESH_PEND	RX3_THRESH_PEND	RX2_THRESH_PEND	RX1_THRESH_PEND	RX0_THRESH_PEND
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

7	6	5	4	3	2	1	0
RX7_PEND	RX6_PEND	RX5_PEND	RX4_PEND	RX3_PEND	RX2_PEND	RX1_PEND	RX0_PEND
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-45. RX_INTSTAT_RAW Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RX7_THRESH_PEND	R	0h	RX7_THRESH_PEND raw int read (before mask).
14	RX6_THRESH_PEND	R	0h	RX6_THRESH_PEND raw int read (before mask).
13	RX5_THRESH_PEND	R	0h	RX5_THRESH_PEND raw int read (before mask).
12	RX4_THRESH_PEND	R	0h	RX4_THRESH_PEND raw int read (before mask).
11	RX3_THRESH_PEND	R	0h	RX3_THRESH_PEND raw int read (before mask).
10	RX2_THRESH_PEND	R	0h	RX2_THRESH_PEND raw int read (before mask).
9	RX1_THRESH_PEND	R	0h	RX1_THRESH_PEND raw int read (before mask).
8	RX0_THRESH_PEND	R	0h	RX0_THRESH_PEND raw int read (before mask).
7	RX7_PEND	R	0h	RX7_PEND raw int read (before mask).
6	RX6_PEND	R	0h	RX6_PEND raw int read (before mask).
5	RX5_PEND	R	0h	RX5_PEND raw int read (before mask).
4	RX4_PEND	R	0h	RX4_PEND raw int read (before mask).
3	RX3_PEND	R	0h	RX3_PEND raw int read (before mask).
2	RX2_PEND	R	0h	RX2_PEND raw int read (before mask).
1	RX1_PEND	R	0h	RX1_PEND raw int read (before mask).
0	RX0_PEND	R	0h	RX0_PEND raw int read (before mask).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.10 RX_INTSTAT_MASKED Register (offset = A4h) [reset = 0h]

CPDMA_INT RX INTERRUPT STATUS REGISTER (MASKED VALUE)

RX_INTSTAT_MASKED is shown in Figure 13-38 and described in Table 13-46.

Figure 13-38. RX_INTSTAT_MASKED Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
RX7_THRESH_PEND	RX6_THRESH_PEND	RX5_THRESH_PEND	RX4_THRESH_PEND	RX3_THRESH_PEND	RX2_THRESH_PEND	RX1_THRESH_PEND	RX0_THRESH_PEND
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

7	6	5	4	3	2	1	0
RX7_PEND	RX6_PEND	RX5_PEND	RX4_PEND	RX3_PEND	RX2_PEND	RX1_PEND	RX0_PEND
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-46. RX_INTSTAT_MASKED Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RX7_THRESH_PEND	R	0h	RX7_THRESH_PEND masked int read.
14	RX6_THRESH_PEND	R	0h	RX6_THRESH_PEND masked int read.
13	RX5_THRESH_PEND	R	0h	RX5_THRESH_PEND masked int read.
12	RX4_THRESH_PEND	R	0h	RX4_THRESH_PEND masked int read.
11	RX3_THRESH_PEND	R	0h	RX3_THRESH_PEND masked int read.
10	RX2_THRESH_PEND	R	0h	RX2_THRESH_PEND masked int read.
9	RX1_THRESH_PEND	R	0h	RX1_THRESH_PEND masked int read.
8	RX0_THRESH_PEND	R	0h	RX0_THRESH_PEND masked int read.
7	RX7_PEND	R	0h	RX7_PEND masked int read.
6	RX6_PEND	R	0h	RX6_PEND masked int read.
5	RX5_PEND	R	0h	RX5_PEND masked int read.
4	RX4_PEND	R	0h	RX4_PEND masked int read.
3	RX3_PEND	R	0h	RX3_PEND masked int read.
2	RX2_PEND	R	0h	RX2_PEND masked int read.
1	RX1_PEND	R	0h	RX1_PEND masked int read.
0	RX0_PEND	R	0h	RX0_PEND masked int read.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.11 RX_INTMASK_SET Register (offset = A8h) [reset = 0h]

CPDMA_INT RX INTERRUPT MASK SET REGISTER

RX_INTMASK_SET is shown in Figure 13-39 and described in Table 13-47.

Figure 13-39. RX_INTMASK_SET Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
RX7_THRESH_PEND_MASK	RX6_THRESH_PEND_MASK	RX5_THRESH_PEND_MASK	RX4_THRESH_PEND_MASK	RX3_THRESH_PEND_MASK	RX2_THRESH_PEND_MASK	RX1_THRESH_PEND_MASK	RX0_THRESH_PEND_MASK
W-0h	R-0h	W-0h	R-0h	W-0h	R-0h	W-0h	R-0h

7	6	5	4	3	2	1	0
RX7_PEND_MASK	RX6_PEND_MASK	RX5_PEND_MASK	RX4_PEND_MASK	RX3_PEND_MASK	RX2_PEND_MASK	RX1_PEND_MASK	RX0_PEND_MASK
W-0h	R-0h	W-0h	R-0h	W-0h	R-0h	W-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-47. RX_INTMASK_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RX7_THRESH_PEND_MASK	W	0h	RX Channel 7 Threshold Pending Int. Mask - Write one to enable Int.
14	RX6_THRESH_PEND_MASK	R	0h	RX Channel 6 Threshold Pending Int. Mask - Write one to enable Int.
13	RX5_THRESH_PEND_MASK	W	0h	RX Channel 5 Threshold Pending Int. Mask - Write one to enable Int.
12	RX4_THRESH_PEND_MASK	R	0h	RX Channel 4 Threshold Pending Int. Mask - Write one to enable Int.
11	RX3_THRESH_PEND_MASK	W	0h	RX Channel 3 Threshold Pending Int. Mask - Write one to enable Int.
10	RX2_THRESH_PEND_MASK	R	0h	RX Channel 2 Threshold Pending Int. Mask - Write one to enable Int.
9	RX1_THRESH_PEND_MASK	W	0h	RX Channel 1 Threshold Pending Int. Mask - Write one to enable Int.
8	RX0_THRESH_PEND_MASK	R	0h	RX Channel 0 Threshold Pending Int. Mask - Write one to enable Int.
7	RX7_PEND_MASK	W	0h	RX Channel 7 Pending Int. Mask - Write one to enable Int.
6	RX6_PEND_MASK	R	0h	RX Channel 6 Pending Int. Mask - Write one to enable Int.
5	RX5_PEND_MASK	W	0h	RX Channel 5 Pending Int. Mask - Write one to enable Int.
4	RX4_PEND_MASK	R	0h	RX Channel 4 Pending Int. Mask - Write one to enable Int.
3	RX3_PEND_MASK	W	0h	RX Channel 3 Pending Int. Mask - Write one to enable Int.
2	RX2_PEND_MASK	R	0h	RX Channel 2 Pending Int. Mask - Write one to enable Int.
1	RX1_PEND_MASK	W	0h	RX Channel 1 Pending Int. Mask - Write one to enable Int.
0	RX0_PEND_MASK	R	0h	RX Channel 0 Pending Int. Mask - Write one to enable Int.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.12 RX_INTMASK_CLEAR Register (offset = ACh) [reset = 0h]

CPDMA_INT RX INTERRUPT MASK CLEAR REGISTER

RX_INTMASK_CLEAR is shown in Figure 13-40 and described in Table 13-48.

Figure 13-40. RX_INTMASK_CLEAR Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
RX7_THRESH_PEND_MASK	RX6_THRESH_PEND_MASK	RX5_THRESH_PEND_MASK	RX4_THRESH_PEND_MASK	RX3_THRESH_PEND_MASK	RX2_THRESH_PEND_MASK	RX1_THRESH_PEND_MASK	RX0_THRESH_PEND_MASK
W-0h	R-0h	W-0h	R-0h	W-0h	R-0h	W-0h	R-0h

7	6	5	4	3	2	1	0
RX7_PEND_MASK	RX6_PEND_MASK	RX5_PEND_MASK	RX4_PEND_MASK	RX3_PEND_MASK	RX2_PEND_MASK	RX1_PEND_MASK	RX0_PEND_MASK
W-0h	R-0h	W-0h	R-0h	W-0h	R-0h	W-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-48. RX_INTMASK_CLEAR Register Field Descriptions

Bit	Field	Type	Reset	Description
15	RX7_THRESH_PEND_MASK	W	0h	RX Channel 7 Threshold Pending Int. Mask - Write one to disable Int.
14	RX6_THRESH_PEND_MASK	R	0h	RX Channel 6 Threshold Pending Int. Mask - Write one to disable Int.
13	RX5_THRESH_PEND_MASK	W	0h	RX Channel 5 Threshold Pending Int. Mask - Write one to disable Int.
12	RX4_THRESH_PEND_MASK	R	0h	RX Channel 4 Threshold Pending Int. Mask - Write one to disable Int.
11	RX3_THRESH_PEND_MASK	W	0h	RX Channel 3 Threshold Pending Int. Mask - Write one to disable Int.
10	RX2_THRESH_PEND_MASK	R	0h	RX Channel 2 Threshold Pending Int. Mask - Write one to disable Int.
9	RX1_THRESH_PEND_MASK	W	0h	RX Channel 1 Threshold Pending Int. Mask - Write one to disable Int.
8	RX0_THRESH_PEND_MASK	R	0h	RX Channel 0 Threshold Pending Int. Mask - Write one to disable Int.
7	RX7_PEND_MASK	W	0h	RX Channel 7 Pending Int. Mask - Write one to disable Int.
6	RX6_PEND_MASK	R	0h	RX Channel 6 Pending Int. Mask - Write one to disable Int.
5	RX5_PEND_MASK	W	0h	RX Channel 5 Pending Int. Mask - Write one to disable Int.
4	RX4_PEND_MASK	R	0h	RX Channel 4 Pending Int. Mask - Write one to disable Int.
3	RX3_PEND_MASK	W	0h	RX Channel 3 Pending Int. Mask - Write one to disable Int.
2	RX2_PEND_MASK	R	0h	RX Channel 2 Pending Int. Mask - Write one to disable Int.
1	RX1_PEND_MASK	W	0h	RX Channel 1 Pending Int. Mask - Write one to disable Int.
0	RX0_PEND_MASK	R	0h	RX Channel 0 Pending Int. Mask - Write one to disable Int.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.13 DMA_INTSTAT_RAW Register (offset = B0h) [reset = 0h]

CPDMA_INT DMA INTERRUPT STATUS REGISTER (RAW VALUE)

DMA_INTSTAT_RAW is shown in Figure 13-41 and described in Table 13-49.

Figure 13-41. DMA_INTSTAT_RAW Register

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						HOST_PEND	STAT_PEND
						R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-49. DMA_INTSTAT_RAW Register Field Descriptions

Bit	Field	Type	Reset	Description
1	HOST_PEND	R	0h	Host Pending Interrupt - raw int read (before mask).
0	STAT_PEND	R	0h	Statistics Pending Interrupt - raw int read (before mask).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.14 DMA_INTSTAT_MASKED Register (offset = B4h) [reset = 0h]

CPDMA_INT DMA INTERRUPT STATUS REGISTER (MASKED VALUE)

DMA_INTSTAT_MASKED is shown in Figure 13-42 and described in Table 13-50.

Figure 13-42. DMA_INTSTAT_MASKED Register

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						HOST_PEND	STAT_PEND
						R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-50. DMA_INTSTAT_MASKED Register Field Descriptions

Bit	Field	Type	Reset	Description
1	HOST_PEND	R	0h	Host Pending Interrupt - masked interrupt read.
0	STAT_PEND	R	0h	Statistics Pending Interrupt - masked interrupt read.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.15 DMA_INTMASK_SET Register (offset = B8h) [reset = 0h]

CPDMA_INT DMA INTERRUPT MASK SET REGISTER

DMA_INTMASK_SET is shown in Figure 13-43 and described in Table 13-51.

Figure 13-43. DMA_INTMASK_SET Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved						HOST_ERR_INT_MASK	STAT_INT_MASK
						W-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-51. DMA_INTMASK_SET Register Field Descriptions

Bit	Field	Type	Reset	Description
1	HOST_ERR_INT_MASK	W	0h	Host Error Interrupt Mask - Write one to enable interrupt.
0	STAT_INT_MASK	R	0h	Statistics Interrupt Mask - Write one to enable interrupt.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.16 DMA_INTMASK_CLEAR Register (offset = BCh) [reset = 0h]

CPDMA_INT DMA INTERRUPT MASK CLEAR REGISTER

DMA_INTMASK_CLEAR is shown in Figure 13-44 and described in Table 13-52.

Figure 13-44. DMA_INTMASK_CLEAR Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved						HOST_ERR_INT_MASK	STAT_INT_MASK
						W-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-52. DMA_INTMASK_CLEAR Register Field Descriptions

Bit	Field	Type	Reset	Description
1	HOST_ERR_INT_MASK	W	0h	Host Error Interrupt Mask - Write one to disable interrupt.
0	STAT_INT_MASK	R	0h	Statistics Interrupt Mask - Write one to disable interrupt.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.17 RX0_PENDTHRESH Register (offset = C0h) [reset = 0h]

CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 0

RX0_PENDTHRESH is shown in Figure 13-45 and described in Table 13-53.

Figure 13-45. RX0_PENDTHRESH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								RX_PENDTHRESH							
																								R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-53. RX0_PENDTHRESH Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	RX_PENDTHRESH	R/W	0h	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.18 RX1_PENDTHRESH Register (offset = C4h) [reset = 0h]

CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 1

RX1_PENDTHRESH is shown in Figure 13-46 and described in Table 13-54.

Figure 13-46. RX1_PENDTHRESH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								RX_PENDTHRESH							
																								R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-54. RX1_PENDTHRESH Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	RX_PENDTHRESH	R/W	0h	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.19 RX_CONTROL Register (offset = 14h) [reset = 0h]

CPDMA_REGS RX CONTROL REGISTER

RX_CONTROL is shown in Figure 13-47 and described in Table 13-55.

Figure 13-47. RX_CONTROL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															RX_EN
																															R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-55. RX_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
0	RX_EN	R/W	0h	RX DMA Enable 0 - Disabled 1 - Enabled

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.20 RX2_PENDTHRESH Register (offset = C8h) [reset = 0h]

CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 2

RX2_PENDTHRESH is shown in Figure 13-48 and described in Table 13-56.

Figure 13-48. RX2_PENDTHRESH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								RX_PENDTHRESH							
																								R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-56. RX2_PENDTHRESH Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	RX_PENDTHRESH	R/W	0h	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.21 RX3_PENDTHRESH Register (offset = CCh) [reset = 0h]

CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 3

RX3_PENDTHRESH is shown in Figure 13-49 and described in Table 13-57.

Figure 13-49. RX3_PENDTHRESH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								RX_PENDTHRESH							
																								R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-57. RX3_PENDTHRESH Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	RX_PENDTHRESH	R/W	0h	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.22 RX4_PENDTHRESH Register (offset = D0h) [reset = 0h]

CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 4

RX4_PENDTHRESH is shown in Figure 13-50 and described in Table 13-58.

Figure 13-50. RX4_PENDTHRESH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								RX_PENDTHRESH							
																								R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-58. RX4_PENDTHRESH Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	RX_PENDTHRESH	R/W	0h	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.23 RX5_PENDTHRESH Register (offset = D4h) [reset = 0h]

CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 5

RX5_PENDTHRESH is shown in Figure 13-51 and described in Table 13-59.

Figure 13-51. RX5_PENDTHRESH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								RX_PENDTHRESH							
																								R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-59. RX5_PENDTHRESH Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	RX_PENDTHRESH	R/W	0h	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.24 RX6_PENDTHRESH Register (offset = D8h) [reset = 0h]

CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 6

RX6_PENDTHRESH is shown in Figure 13-52 and described in Table 13-60.

Figure 13-52. RX6_PENDTHRESH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								RX_PENDTHRESH							
																								R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-60. RX6_PENDTHRESH Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	RX_PENDTHRESH	R/W	0h	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.25 RX7_PENDTHRESH Register (offset = DCh) [reset = 0h]

CPDMA_INT RECEIVE THRESHOLD PENDING REGISTER CHANNEL 7

RX7_PENDTHRESH is shown in Figure 13-53 and described in Table 13-61.

Figure 13-53. RX7_PENDTHRESH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								RX_PENDTHRESH							
																								R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-61. RX7_PENDTHRESH Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	RX_PENDTHRESH	R/W	0h	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.26 RX0_FREEBUFFER Register (offset = E0h) [reset = 0h]

CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 0

RX0_FREEBUFFER is shown in Figure 13-54 and described in Table 13-62.

Figure 13-54. RX0_FREEBUFFER Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																RX_FREEBUFFER															
																						W-0h									

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-62. RX0_FREEBUFFER Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RX_FREEBUFFER	W	0h	Rx Free Buffer Count - This field contains the count of free buffers available. The rx_pendthresh value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.27 RX1_FREEBUFFER Register (offset = E4h) [reset = 0h]

CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 1

RX1_FREEBUFFER is shown in Figure 13-55 and described in Table 13-63.

Figure 13-55. RX1_FREEBUFFER Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																RX_FREEBUFFER															
																						W-0h									

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-63. RX1_FREEBUFFER Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RX_FREEBUFFER	W	0h	Rx Free Buffer Count - This field contains the count of free buffers available. The rx_pendthresh value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.28 RX2_FREEBUFFER Register (offset = E8h) [reset = 0h]

CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 2

RX2_FREEBUFFER is shown in Figure 13-56 and described in Table 13-64.

Figure 13-56. RX2_FREEBUFFER Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																RX_FREEBUFFER															
																						W-0h									

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-64. RX2_FREEBUFFER Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RX_FREEBUFFER	W	0h	Rx Free Buffer Count - This field contains the count of free buffers available. The rx_pendthresh value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.29 RX3_FREEBUFFER Register (offset = ECh) [reset = 0h]

CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 3

RX3_FREEBUFFER is shown in Figure 13-57 and described in Table 13-65.

Figure 13-57. RX3_FREEBUFFER Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																RX_FREEBUFFER															
																						W-0h									

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-65. RX3_FREEBUFFER Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RX_FREEBUFFER	W	0h	Rx Free Buffer Count - This field contains the count of free buffers available. The rx_pendthresh value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.30 RX_TEARDOWN Register (offset = 18h) [reset = 0h]

CPDMA_REGS RX TEARDOWN REGISTER

RX_TEARDOWN is shown in Figure 13-58 and described in Table 13-66.

Figure 13-58. RX_TEARDOWN Register

31	30	29	28	27	26	25	24
RX_TDN_RDY	Reserved						
R-0h							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved					RX_TDN_CH		
						R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-66. RX_TEARDOWN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	RX_TDN_RDY	R	0h	Teardown Ready - read as zero, but is always assumed to be one (unused).
2-0	RX_TDN_CH	R/W	0h	Rx Teardown Channel -Receive channel teardown is commanded by writing the encoded value of the receive channel to be torn down. The teardown register is read as zero. 000 - teardown receive channel 0 ... 111 - teardown receive channel 7

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.31 RX4_FREEBUFFER Register (offset = F0h) [reset = 0h]

CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 4

RX4_FREEBUFFER is shown in Figure 13-59 and described in Table 13-67.

Figure 13-59. RX4_FREEBUFFER Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																RX_FREEBUFFER															
																							W-0h								

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-67. RX4_FREEBUFFER Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RX_FREEBUFFER	W	0h	Rx Free Buffer Count - This field contains the count of free buffers available. The rx_pendthresh value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.32 RX5_FREEBUFFER Register (offset = F4h) [reset = 0h]

CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 5

RX5_FREEBUFFER is shown in Figure 13-60 and described in Table 13-68.

Figure 13-60. RX5_FREEBUFFER Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																RX_FREEBUFFER															
																						W-0h									

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-68. RX5_FREEBUFFER Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RX_FREEBUFFER	W	0h	Rx Free Buffer Count - This field contains the count of free buffers available. The rx_pendthresh value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.33 RX6_FREEBUFFER Register (offset = F8h) [reset = 0h]

CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 6

RX6_FREEBUFFER is shown in Figure 13-61 and described in Table 13-69.

Figure 13-61. RX6_FREEBUFFER Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																RX_FREEBUFFER															
																						W-0h									

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-69. RX6_FREEBUFFER Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RX_FREEBUFFER	W	0h	Rx Free Buffer Count - This field contains the count of free buffers available. The rx_pendthresh value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.34 RX7_FREEBUFFER Register (offset = FCh) [reset = 0h]

CPDMA_INT RECEIVE FREE BUFFER REGISTER CHANNEL 7

RX7_FREEBUFFER is shown in Figure 13-62 and described in Table 13-70.

Figure 13-62. RX7_FREEBUFFER Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																RX_FREEBUFFER															
																						W-0h									

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-70. RX7_FREEBUFFER Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RX_FREEBUFFER	W	0h	Rx Free Buffer Count - This field contains the count of free buffers available. The rx_pendthresh value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.35 TX0_HDP Register (offset = A00h) [reset = 0h]

CPDMA_STATERAM TX CHANNEL 0 HEAD DESC POINTER *

TX0_HDP is shown in Figure 13-63 and described in Table 13-71.

Figure 13-63. TX0_HDP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_HDP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-71. TX0_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_HDP	R/W	0h	TX Channel (0. 7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.36 TX1_HDP Register (offset = A04h) [reset = 0h]

CPDMA_STATERAM TX CHANNEL 1 HEAD DESC POINTER *

TX1_HDP is shown in Figure 13-64 and described in Table 13-72.

Figure 13-64. TX1_HDP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_HDP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-72. TX1_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_HDP	R/W	0h	TX Channel (0. 7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.37 TX2_HDP Register (offset = A08h) [reset = 0h]

CPDMA_STATERAM TX CHANNEL 2 HEAD DESC POINTER *

TX2_HDP is shown in Figure 13-65 and described in Table 13-73.

Figure 13-65. TX2_HDP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_HDP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-73. TX2_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_HDP	R/W	0h	TX Channel (0. 7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.38 TX3_HDP Register (offset = A0Ch) [reset = 0h]

CPDMA_STATERAM TX CHANNEL 3 HEAD DESC POINTER *

TX3_HDP is shown in Figure 13-66 and described in Table 13-74.

Figure 13-66. TX3_HDP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_HDP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-74. TX3_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_HDP	R/W	0h	TX Channel (0. 7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.39 TX4_HDP Register (offset = A10h) [reset = 0h]

CPDMA_STATERAM TX CHANNEL 4 HEAD DESC POINTER *

TX4_HDP is shown in Figure 13-67 and described in Table 13-75.

Figure 13-67. TX4_HDP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_HDP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-75. TX4_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_HDP	R/W	0h	TX Channel (0. 7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.40 TX5_HDP Register (offset = A14h) [reset = 0h]

CPDMA_STATERAM TX CHANNEL 5 HEAD DESC POINTER *

TX5_HDP is shown in Figure 13-68 and described in Table 13-76.

Figure 13-68. TX5_HDP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_HDP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-76. TX5_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_HDP	R/W	0h	TX Channel (0. 7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.41 TX6_HDP Register (offset = A18h) [reset = 0h]

CPDMA_STATERAM TX CHANNEL 6 HEAD DESC POINTER *

TX6_HDP is shown in Figure 13-69 and described in Table 13-77.

Figure 13-69. TX6_HDP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_HDP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-77. TX6_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_HDP	R/W	0h	TX Channel (0. 7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.42 TX7_HDP Register (offset = A1Ch) [reset = 0h]

CPDMA_STATERAM TX CHANNEL 7 HEAD DESC POINTER *

TX7_HDP is shown in Figure 13-70 and described in Table 13-78.

Figure 13-70. TX7_HDP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_HDP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-78. TX7_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_HDP	R/W	0h	TX Channel (0. 7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.43 RX0_HDP Register (offset = A20h) [reset = 0h]

CPDMA_STATERAM RX 0 CHANNEL 0 HEAD DESC POINTER *

RX0_HDP is shown in Figure 13-71 and described in Table 13-79.

Figure 13-71. RX0_HDP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_HDP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-79. RX0_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_HDP	R/W	0h	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.44 RX1_HDP Register (offset = A24h) [reset = 0h]

CPDMA_STATERAM RX 1 CHANNEL 1 HEAD DESC POINTER *

RX1_HDP is shown in Figure 13-72 and described in Table 13-80.

Figure 13-72. RX1_HDP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_HDP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-80. RX1_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_HDP	R/W	0h	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.45 RX2_HDP Register (offset = A28h) [reset = 0h]

CPDMA_STATERAM RX 2 CHANNEL 2 HEAD DESC POINTER *

RX2_HDP is shown in Figure 13-73 and described in Table 13-81.

Figure 13-73. RX2_HDP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_HDP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-81. RX2_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_HDP	R/W	0h	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.46 RX3_HDP Register (offset = A2Ch) [reset = 0h]

CPDMA_STATERAM RX 3 CHANNEL 3 HEAD DESC POINTER *

RX3_HDP is shown in Figure 13-74 and described in Table 13-82.

Figure 13-74. RX3_HDP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_HDP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-82. RX3_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_HDP	R/W	0h	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.47 RX4_HDP Register (offset = A30h) [reset = 0h]

CPDMA_STATERAM RX 4 CHANNEL 4 HEAD DESC POINTER *

RX4_HDP is shown in Figure 13-75 and described in Table 13-83.

Figure 13-75. RX4_HDP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_HDP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-83. RX4_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_HDP	R/W	0h	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.48 RX5_HDP Register (offset = A34h) [reset = 0h]

CPDMA_STATERAM RX 5 CHANNEL 5 HEAD DESC POINTER *

RX5_HDP is shown in Figure 13-76 and described in Table 13-84.

Figure 13-76. RX5_HDP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_HDP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-84. RX5_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_HDP	R/W	0h	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.49 RX6_HDP Register (offset = A38h) [reset = 0h]

CPDMA_STATERAM RX 6 CHANNEL 6 HEAD DESC POINTER *

RX6_HDP is shown in Figure 13-77 and described in Table 13-85.

Figure 13-77. RX6_HDP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_HDP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-85. RX6_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_HDP	R/W	0h	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.50 RX7_HDP Register (offset = A3Ch) [reset = 0h]

CPDMA_STATERAM RX 7 CHANNEL 7 HEAD DESC POINTER *

RX7_HDP is shown in Figure 13-78 and described in Table 13-86.

Figure 13-78. RX7_HDP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_HDP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-86. RX7_HDP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_HDP	R/W	0h	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.51 TX0_CP Register (offset = A40h) [reset = 0h]

CPDMA_STATERAM TX CHANNEL 0 COMPLETION POINTER REGISTER

TX0_CP is shown in Figure 13-79 and described in Table 13-87.

Figure 13-79. TX0_CP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_CP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-87. TX0_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_CP	R/W	0h	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.52 TX1_CP Register (offset = A44h) [reset = 0h]

CPDMA_STATERAM TX CHANNEL 1 COMPLETION POINTER REGISTER *

TX1_CP is shown in Figure 13-80 and described in Table 13-88.

Figure 13-80. TX1_CP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_CP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-88. TX1_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_CP	R/W	0h	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.53 TX2_CP Register (offset = A48h) [reset = 0h]

CPDMA_STATERAM TX CHANNEL 2 COMPLETION POINTER REGISTER *

TX2_CP is shown in Figure 13-81 and described in Table 13-89.

Figure 13-81. TX2_CP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_CP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-89. TX2_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_CP	R/W	0h	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.54 TX3_CP Register (offset = A4Ch) [reset = 0h]

CPDMA_STATERAM TX CHANNEL 3 COMPLETION POINTER REGISTER *

TX3_CP is shown in Figure 13-82 and described in Table 13-90.

Figure 13-82. TX3_CP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_CP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-90. TX3_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_CP	R/W	0h	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.55 TX4_CP Register (offset = A50h) [reset = 0h]

CPDMA_STATERAM TX CHANNEL 4 COMPLETION POINTER REGISTER *

TX4_CP is shown in Figure 13-83 and described in Table 13-91.

Figure 13-83. TX4_CP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_CP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-91. TX4_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_CP	R/W	0h	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.56 TX5_CP Register (offset = A54h) [reset = 0h]

CPDMA_STATERAM TX CHANNEL 5 COMPLETION POINTER REGISTER *

TX5_CP is shown in Figure 13-84 and described in Table 13-92.

Figure 13-84. TX5_CP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_CP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-92. TX5_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_CP	R/W	0h	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.57 TX6_CP Register (offset = A58h) [reset = 0h]

CPDMA_STATERAM TX CHANNEL 6 COMPLETION POINTER REGISTER *

TX6_CP is shown in Figure 13-85 and described in Table 13-93.

Figure 13-85. TX6_CP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_CP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-93. TX6_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_CP	R/W	0h	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.58 TX7_CP Register (offset = A5Ch) [reset = 0h]

CPDMA_STATERAM TX CHANNEL 7 COMPLETION POINTER REGISTER *

TX7_CP is shown in Figure 13-86 and described in Table 13-94.

Figure 13-86. TX7_CP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_CP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-94. TX7_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TX_CP	R/W	0h	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.59 RX0_CP Register (offset = A60h) [reset = 0h]

CPDMA_STATERAM RX CHANNEL 0 COMPLETION POINTER REGISTER *

RX0_CP is shown in Figure 13-87 and described in Table 13-95.

Figure 13-87. RX0_CP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_CP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-95. RX0_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_CP	R/W	0h	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.60 RX1_CP Register (offset = A64h) [reset = 0h]

CPDMA_STATERAM RX CHANNEL 1 COMPLETION POINTER REGISTER *

RX1_CP is shown in Figure 13-88 and described in Table 13-96.

Figure 13-88. RX1_CP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_CP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-96. RX1_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_CP	R/W	0h	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.61 RX2_CP Register (offset = A68h) [reset = 0h]

CPDMA_STATERAM RX CHANNEL 2 COMPLETION POINTER REGISTER *

RX2_CP is shown in Figure 13-89 and described in Table 13-97.

Figure 13-89. RX2_CP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_CP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-97. RX2_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_CP	R/W	0h	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.62 RX3_CP Register (offset = A6Ch) [reset = 0h]

CPDMA_STATERAM RX CHANNEL 3 COMPLETION POINTER REGISTER *

RX3_CP is shown in Figure 13-90 and described in Table 13-98.

Figure 13-90. RX3_CP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_CP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-98. RX3_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_CP	R/W	0h	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.63 RX4_CP Register (offset = A70h) [reset = 0h]

CPDMA_STATERAM RX CHANNEL 4 COMPLETION POINTER REGISTER *

RX4_CP is shown in Figure 13-91 and described in Table 13-99.

Figure 13-91. RX4_CP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_CP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-99. RX4_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_CP	R/W	0h	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.64 RX5_CP Register (offset = A74h) [reset = 0h]

CPDMA_STATERAM RX CHANNEL 5 COMPLETION POINTER REGISTER *

RX5_CP is shown in Figure 13-92 and described in Table 13-100.

Figure 13-92. RX5_CP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_CP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-100. RX5_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_CP	R/W	0h	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.65 RX6_CP Register (offset = A78h) [reset = 0h]

CPDMA_STATERAM RX CHANNEL 6 COMPLETION POINTER REGISTER *

RX6_CP is shown in Figure 13-93 and described in Table 13-101.

Figure 13-93. RX6_CP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_CP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-101. RX6_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_CP	R/W	0h	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.66 RX7_CP Register (offset = A7Ch) [reset = 0h]

CPDMA_STATERAM RX CHANNEL 7 COMPLETION POINTER REGISTER *

RX7_CP is shown in Figure 13-94 and described in Table 13-102.

Figure 13-94. RX7_CP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_CP																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-102. RX7_CP Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	RX_CP	R/W	0h	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.67 CPDMA_SOFT_RESET Register (offset = 1Ch) [reset = 0h]

CPDMA_REGS SOFT RESET REGISTER

CPDMA_SOFT_RESET is shown in Figure 13-95 and described in Table 13-103.

Figure 13-95. CPDMA_SOFT_RESET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															SOFT_RESET
																															R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-103. CPDMA_SOFT_RESET Register Field Descriptions

Bit	Field	Type	Reset	Description
0	SOFT_RESET	R/W	0h	Software reset - Writing a one to this bit causes the CPDMA logic to be reset. Software reset occurs when the RX and TX DMA Controllers are in an idle state to avoid locking up the VBUSP bus. After writing a one to this bit, it may be polled to determine if the reset has occurred. If a one is read, the reset has not yet occurred. If a zero is read then reset has occurred.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.68 DMACONTROL Register (offset = 20h) [reset = 0h]

CPDMA_REGS CPDMA CONTROL REGISTER

DMACONTROL is shown in Figure 13-96 and described in Table 13-104.

Figure 13-96. DMACONTROL Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
TX_RLIM							
R/W-0h							

7	6	5	4	3	2	1	0
Reserved			RX_CEF	CMD_IDLE	RX_OFFLEN_BLOCK	RX_OWNERSHIP	TX_PTYPE
			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-104. DMACONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	TX_RLIM	R/W	0h	Transmit Rate Limit Channel Bus 00000000 - no rate-limited channels 10000000 - channel 7 is rate-limited 11000000 - channels 7 downto 6 are rate-limited 11100000 - channels 7 downto 5 are rate-limited 11110000 - channels 7 downto 4 are rate-limited 11111000 - channels 7 downto 3 are rate-limited 11111100 - channels 7 downto 2 are rate-limited 11111110 - channels 7 downto 1 are rate-limited 11111111 - channels 7 downto 0 are rate-limited all others invalid - this bus must be set msb towards lsb. tx_ptype must be set if any tx_rlim bit is set for fixed priority.
4	RX_CEF	R/W	0h	RX Copy Error Frames Enable - Enables DMA overrun frames to be transferred to memory (up to the point of overrun). The overrun error bit will be set in the frame EOP buffer descriptor. Overrun frame data will be filtered when rx_cef is not set. Frames coming from the receive FIFO with other error bits set are not effected by this bit. 0 - Frames containing overrun errors are filtered. 1 - Frames containing overrun errors are transferred to memory.
3	CMD_IDLE	R/W	0h	Command Idle 0 - Idle not commanded 1 - Idle Commanded (read idle in DMAStatus)
2	RX_OFFLEN_BLOCK	R/W	0h	Receive Offset/Length word write block. 0 - Do not block the DMA writes to the receive buffer descriptor offset/buffer length word. The offset/buffer length word is written as specified in CPPI 3.0. 1 - Block all CPDMA DMA controller writes to the receive buffer descriptor offset/buffer length words during CPPI packet processing. when this bit is set, the CPDMA will never write the third word to any receive buffer descriptor.
1	RX_OWNERSHIP	R/W	0h	Receive Ownership Write Bit Value. 0 - The CPDMA writes the receive ownership bit to zero at the end of packet processing as specified in CPPI 3.0. 1 - The CPDMA writes the receive ownership bit to one at the end of packet processing. Users who do not use the ownership mechanism can use this mode to preclude the necessity of software having to set this bit each time the buffer descriptor is used.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 13-104. DMACONTROL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
0	TX_PTYPE	R/W	0h	Transmit Queue Priority Type 0 - The queue uses a round robin scheme to select the next channel for transmission. 1 - The queue uses a fixed (channel 7 highest priority) priority scheme to select the next channel for transmission

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.69 DMASTATUS Register (offset = 24h) [reset = 0h]

CPDMA_REGS CPDMA STATUS REGISTER

DMASTATUS is shown in Figure 13-97 and described in Table 13-105.

Figure 13-97. DMASTATUS Register

31	30	29	28	27	26	25	24
IDLE	Reserved						
R-0h							

23	22	21	20	19	18	17	16
TX_HOST_ERR_CODE				Reserved	TX_ERR_CH		
R-0h					R-0h		

15	14	13	12	11	10	9	8
RX_HOST_ERR_CODE				Reserved	RX_ERR_CH		
R-0h					R-0h		

7	6	5	4	3	2	1	0
Reserved							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-105. DMASTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IDLE	R	0h	Idle Status Bit - Indicates when set that the CPDMA is not transferring a packet on transmit or receive.
23-20	TX_HOST_ERR_CODE	R	0h	TX Host Error Code - This field is set to indicate CPDMA detected TX DMA related host errors. The host should read this field after a HOST_ERR_INT to determine the error. Host error Interrupts require hardware reset in order to recover. A zero packet length is an error, but it is not detected. 0000 - No error 0001 - SOP error. 0010 - Ownership bit not set in SOP buffer. 0011 - Zero Next Buffer Descriptor Pointer Without EOP 0100 - Zero Buffer Pointer. 0101 - Zero Buffer Length 0110 - Packet Length Error (sum of buffers less than packet length) 0111 - reserved ... 1111 - reserved
18-16	TX_ERR_CH	R	0h	TX Host Error Channel - This field indicates which TX channel (if applicable) the host error occurred on. This field is cleared to zero on a host read. 000 - The host error occurred on TX channel 0 ... 111 - The host error occurred on TX channel 7
15-12	RX_HOST_ERR_CODE	R	0h	RX Host Error Code - This field is set to indicate CPDMA detected RX DMA related host errors. The host should read this field after a HOST_ERR_INT to determine the error. Host error Interrupts require hardware reset in order to recover. 0000 - No error 0001 - reserved 0010 - Ownership bit not set in input buffer. 0011 - reserved 0100 - Zero Buffer Pointer. 0101 - Zero buffer length on non-SOP descriptor 0110 - SOP buffer length not greater than offset ... 1111 - reserved
10-8	RX_ERR_CH	R	0h	RX Host Error Channel - This field indicates which RX channel the host error occurred on. This field is cleared to zero on a host read. 000 - The host error occurred on RX channel 0 ... 111 - The host error occurred on RX channel 7

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.70 TX_CONTROL Register (offset = 4h) [reset = 0h]

CPDMA_REGS TX CONTROL REGISTER

TX_CONTROL is shown in Figure 13-98 and described in Table 13-106.

Figure 13-98. TX_CONTROL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															TX_EN
																															R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-106. TX_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
0	TX_EN	R/W	0h	TX Enable 0 - Disabled 1 - Enabled

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.71 RX_BUFFER_OFFSET Register (offset = 28h) [reset = 0h]

CPDMA_REGS RECEIVE BUFFER OFFSET

RX_BUFFER_OFFSET is shown in Figure 13-99 and described in Table 13-107.

Figure 13-99. RX_BUFFER_OFFSET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																RX_BUFFER_OFFSET															
																						R/W-0h									

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-107. RX_BUFFER_OFFSET Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	RX_BUFFER_OFFSET	R/W	0h	Receive Buffer Offset Value - The rx_buffer_offset will be written by the port into each frame SOP buffer descriptor buffer_offset field. The frame data will begin after the rx_buffer_offset value of bytes. A value of 0x0000 indicates that there are no unused bytes at the beginning of the data and that valid data begins on the first byte of the buffer. A value of 0x000F (decimal 15) indicates that the first 15 bytes of the buffer are to be ignored by the port and that valid buffer data starts on byte 16 of the buffer. This value is used for all channels.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.72 EMCONTROL Register (offset = 2Ch) [reset = 0h]

CPDMA_REGS EMULATION CONTROL

EMCONTROL is shown in Figure 13-180 and described in Table 13-191.

Figure 13-100. EMCONTROL Register

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						SOFT	FREE
						R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-108. EMCONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
1	SOFT	R/W	0h	Emulation Soft Bit
0	FREE	R/W	0h	Emulation Free Bit

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.73 TX_PRI0_RATE Register (offset = 30h) [reset = 0h]

CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 0 RATE

TX_PRI0_RATE is shown in Figure 13-101 and described in Table 13-109.

Figure 13-101. TX_PRI0_RATE Register

31	30	29	28	27	26	25	24
Reserved		PRIN_IDLE_CNT					
				R/W-0h			

23	22	21	20	19	18	17	16
PRIN_IDLE_CNT							
				R/W-0h			

15	14	13	12	11	10	9	8
Reserved		PRIN_SEND_CNT					
				R/W-0h			

7	6	5	4	3	2	1	0
PRIN_SEND_CNT							
				R/W-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-109. TX_PRI0_RATE Register Field Descriptions

Bit	Field	Type	Reset	Description
29-16	PRIN_IDLE_CNT	R/W	0h	Priority (7:0) idle count
13-0	PRIN_SEND_CNT	R/W	0h	Priority (7:0) send count

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.74 TX_PRI1_RATE Register (offset = 34h) [reset = 0h]

CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 1 RATE

TX_PRI1_RATE is shown in Figure 13-102 and described in Table 13-110.

Figure 13-102. TX_PRI1_RATE Register

31	30	29	28	27	26	25	24
Reserved		PRIN_IDLE_CNT					
		R/W-0h					

23	22	21	20	19	18	17	16
PRIN_IDLE_CNT							
R/W-0h							

15	14	13	12	11	10	9	8
Reserved		PRIN_SEND_CNT					
		R/W-0h					

7	6	5	4	3	2	1	0
PRIN_SEND_CNT							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-110. TX_PRI1_RATE Register Field Descriptions

Bit	Field	Type	Reset	Description
29-16	PRIN_IDLE_CNT	R/W	0h	Priority (7:0) idle count
13-0	PRIN_SEND_CNT	R/W	0h	Priority (7:0) send count

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.75 TX_PRI2_RATE Register (offset = 38h) [reset = 0h]

CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 2 RATE

TX_PRI2_RATE is shown in Figure 13-103 and described in Table 13-111.

Figure 13-103. TX_PRI2_RATE Register

31	30	29	28	27	26	25	24
Reserved		PRIN_IDLE_CNT					
				R/W-0h			

23	22	21	20	19	18	17	16
PRIN_IDLE_CNT							
				R/W-0h			

15	14	13	12	11	10	9	8
Reserved		PRIN_SEND_CNT					
				R/W-0h			

7	6	5	4	3	2	1	0
PRIN_SEND_CNT							
				R/W-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-111. TX_PRI2_RATE Register Field Descriptions

Bit	Field	Type	Reset	Description
29-16	PRIN_IDLE_CNT	R/W	0h	Priority (7:0) idle count
13-0	PRIN_SEND_CNT	R/W	0h	Priority (7:0) send count

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.76 TX_PRI3_RATE Register (offset = 3Ch) [reset = 0h]

CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 3 RATE

TX_PRI3_RATE is shown in Figure 13-104 and described in Table 13-112.

Figure 13-104. TX_PRI3_RATE Register

31	30	29	28	27	26	25	24
Reserved		PRIN_IDLE_CNT					
		R/W-0h					

23	22	21	20	19	18	17	16
PRIN_IDLE_CNT							
R/W-0h							

15	14	13	12	11	10	9	8
Reserved		PRIN_SEND_CNT					
		R/W-0h					

7	6	5	4	3	2	1	0
PRIN_SEND_CNT							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-112. TX_PRI3_RATE Register Field Descriptions

Bit	Field	Type	Reset	Description
29-16	PRIN_IDLE_CNT	R/W	0h	Priority (7:0) idle count
13-0	PRIN_SEND_CNT	R/W	0h	Priority (7:0) send count

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.77 TX_PRI4_RATE Register (offset = 40h) [reset = 0h]

CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 4 RATE

TX_PRI4_RATE is shown in Figure 13-105 and described in Table 13-113.

Figure 13-105. TX_PRI4_RATE Register

31	30	29	28	27	26	25	24
Reserved		PRIN_IDLE_CNT					
		R/W-0h					

23	22	21	20	19	18	17	16
PRIN_IDLE_CNT							
R/W-0h							

15	14	13	12	11	10	9	8
Reserved		PRIN_SEND_CNT					
		R/W-0h					

7	6	5	4	3	2	1	0
PRIN_SEND_CNT							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-113. TX_PRI4_RATE Register Field Descriptions

Bit	Field	Type	Reset	Description
29-16	PRIN_IDLE_CNT	R/W	0h	Priority (7:0) idle count
13-0	PRIN_SEND_CNT	R/W	0h	Priority (7:0) send count

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.78 TX_PRI5_RATE Register (offset = 44h) [reset = 0h]

CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 5 RATE

TX_PRI5_RATE is shown in Figure 13-106 and described in Table 13-114.

Figure 13-106. TX_PRI5_RATE Register

31	30	29	28	27	26	25	24
Reserved		PRIN_IDLE_CNT					
		R/W-0h					

23	22	21	20	19	18	17	16
PRIN_IDLE_CNT							
R/W-0h							

15	14	13	12	11	10	9	8
Reserved		PRIN_SEND_CNT					
		R/W-0h					

7	6	5	4	3	2	1	0
PRIN_SEND_CNT							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-114. TX_PRI5_RATE Register Field Descriptions

Bit	Field	Type	Reset	Description
29-16	PRIN_IDLE_CNT	R/W	0h	Priority (7:0) idle count
13-0	PRIN_SEND_CNT	R/W	0h	Priority (7:0) send count

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.79 TX_PRI6_RATE Register (offset = 48h) [reset = 0h]

CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 6 RATE

TX_PRI6_RATE is shown in Figure 13-107 and described in Table 13-115.

Figure 13-107. TX_PRI6_RATE Register

31	30	29	28	27	26	25	24
Reserved		PRIN_IDLE_CNT					
		R/W-0h					

23	22	21	20	19	18	17	16
PRIN_IDLE_CNT							
R/W-0h							

15	14	13	12	11	10	9	8
Reserved		PRIN_SEND_CNT					
		R/W-0h					

7	6	5	4	3	2	1	0
PRIN_SEND_CNT							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-115. TX_PRI6_RATE Register Field Descriptions

Bit	Field	Type	Reset	Description
29-16	PRIN_IDLE_CNT	R/W	0h	Priority (7:0) idle count
13-0	PRIN_SEND_CNT	R/W	0h	Priority (7:0) send count

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.80 TX_PRI7_RATE Register (offset = 4Ch) [reset = 0h]

CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 7 RATE

TX_PRI7_RATE is shown in Figure 13-108 and described in Table 13-116.

Figure 13-108. TX_PRI7_RATE Register

31	30	29	28	27	26	25	24
Reserved		PRIN_IDLE_CNT					
				R/W-0h			

23	22	21	20	19	18	17	16
PRIN_IDLE_CNT							
				R/W-0h			

15	14	13	12	11	10	9	8
Reserved		PRIN_SEND_CNT					
				R/W-0h			

7	6	5	4	3	2	1	0
PRIN_SEND_CNT							
				R/W-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-116. TX_PRI7_RATE Register Field Descriptions

Bit	Field	Type	Reset	Description
29-16	PRIN_IDLE_CNT	R/W	0h	Priority (7:0) idle count
13-0	PRIN_SEND_CNT	R/W	0h	Priority (7:0) send count

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.6.81 TX_TEARDOWN Register (offset = 8h) [reset = 0h]

CPDMA_REGS TX TEARDOWN REGISTER

TX_TEARDOWN is shown in Figure 13-109 and described in Table 13-117.

Figure 13-109. TX_TEARDOWN Register

31	30	29	28	27	26	25	24
TX_TDN_RDY	Reserved						
R-0h							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved					TX_TDN_CH		
						R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-117. TX_TEARDOWN Register Field Descriptions

Bit	Field	Type	Reset	Description
31	TX_TDN_RDY	R	0h	Tx Teardown Ready - read as zero, but is always assumed to be one (unused).
2-0	TX_TDN_CH	R/W	0h	Tx Teardown Channel - Transmit channel teardown is commanded by writing the encoded value of the transmit channel to be torn down. The teardown register is read as zero. 000 - teardown transmit channel 0 ... 111 - teardown transmit channel 7

13.7 CPSW_CPTS Registers

Table 13-118 lists the memory-mapped registers for the CPSW_CPTS. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 13-118 should be considered as reserved locations and the register contents should not be modified.

Table 13-118. CPSW_CPTS REGISTERS

Offset	Acronym	Register Description	Section
h + (0h)	CPTS_IDVER	IDENTIFICATION AND VERSION REGISTER	Section 13.7.1
h + (Ch)	CPTS_TS_PUSH	TIME STAMP EVENT PUSH REGISTER	Section 13.7.2
h + (10h)	CPTS_TS_LOAD_VAL	TIME STAMP LOAD VALUE REGISTER	Section 13.7.3
h + (14h)	CPTS_TS_LOAD_EN	TIME STAMP LOAD ENABLE REGISTER	Section 13.7.4
h + (20h)	CPTS_INTSTAT_RAW	TIME SYNC INTERRUPT STATUS RAW REGISTER	Section 13.7.5
h + (24h)	CPTS_INTSTAT_MASKED	TIME SYNC INTERRUPT STATUS MASKED REGISTER	Section 13.7.6
h + (4h)	CPTS_CONTROL	TIME SYNC CONTROL REGISTER	Section 13.7.7
h + (28h)	CPTS_INT_ENABLE	TIME SYNC INTERRUPT ENABLE REGISTER	Section 13.7.8
h + (30h)	CPTS_EVENT_POP	EVENT INTERRUPT POP REGISTER	Section 13.7.9
h + (34h)	CPTS_EVENT_LOW	LOWER 32-BITS OF THE EVENT VALUE	Section 13.7.10
h + (38h)	CPTS_EVENT_HIGH	UPPER 32-BITS OF THE EVENT VALUE	Section 13.7.11
h + (8h)	CPTS_RFTCLK_SEL	REFERENCE CLOCK SELECT REGISTER	Section 13.7.12

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.7.1 CPTS_IDVER Register (offset = h + (0h)) [reset = 4E8A0101h]

IDENTIFICATION AND VERSION REGISTER

CPTS_IDVER is shown in Figure 13-110 and described in Table 13-119.

Figure 13-110. CPTS_IDVER Register

31	30	29	28	27	26	25	24
TX_IDENT							
R-4E8Ah							
23	22	21	20	19	18	17	16
TX_IDENT							
R-4E8Ah							
15	14	13	12	11	10	9	8
RTL_VER					MAJOR_VER		
R-0h					R-1h		
7	6	5	4	3	2	1	0
MINOR_VER							
R-1h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-119. CPTS_IDVER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	TX_IDENT	R	4E8Ah	TX Identification Value
15-11	RTL_VER	R	0h	RTL Version Value
10-8	MAJOR_VER	R	1h	Major Version Value
7-0	MINOR_VER	R	1h	Minor Version Value

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.7.2 CPTS_TS_PUSH Register (offset = h + (Ch)) [reset = 0h]

TIME STAMP EVENT PUSH REGISTER

CPTS_TS_PUSH is shown in Figure 13-111 and described in Table 13-120.

Figure 13-111. CPTS_TS_PUSH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															TS_PUSH
																															W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-120. CPTS_TS_PUSH Register Field Descriptions

Bit	Field	Type	Reset	Description
0	TS_PUSH	W	0h	Time stamp event push - When a logic high is written to this bit a time stamp event is pushed onto the event FIFO. The time stamp value is the time of the write of this register, not the time of the event read. The time stamp value can then be read on interrupt via the event registers. Software should not push a second time stamp event onto the event FIFO until the first time stamp value has been read from the event FIFO (there should be only one time stamp event in the event FIFO at any given time). This bit is write only and always reads zero.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.7.3 CPTS_TS_LOAD_VAL Register (offset = h + (10h)) [reset = 0h]

TIME STAMP LOAD VALUE REGISTER

CPTS_TS_LOAD_VAL is shown in Figure 13-112 and described in Table 13-121.

Figure 13-112. CPTS_TS_LOAD_VAL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS_LOAD_VAL																															
R/W-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-121. CPTS_TS_LOAD_VAL Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TS_LOAD_VAL	R/W	0h	Time Stamp Load Value - Writing the ts_load_en bit causes the value contained in this register to be written into the time stamp. The time stamp value is read by initiating a time stamp push event, not by reading this register. When reading this register, the value read is not the time stamp, but is the value that was last written to this register.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.7.4 CPTS_TS_LOAD_EN Register (offset = h + (14h)) [reset = 0h]

TIME STAMP LOAD ENABLE REGISTER

CPTS_TS_LOAD_EN is shown in Figure 13-113 and described in Table 13-122.

Figure 13-113. CPTS_TS_LOAD_EN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															TS_LOAD_EN
																															W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-122. CPTS_TS_LOAD_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
0	TS_LOAD_EN	W	0h	Time Stamp Load - Writing a one to this bit enables the time stamp value to be written via the ts_load_val[31:0] register. This feature is included for test purposes. This bit is write only.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.7.5 CPTS_INTSTAT_RAW Register (offset = h + (20h)) [reset = 0h]

TIME SYNC INTERRUPT STATUS RAW REGISTER

CPTS_INTSTAT_RAW is shown in Figure 13-114 and described in Table 13-123.

Figure 13-114. CPTS_INTSTAT_RAW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															TS_PEND_RAW
																															R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-123. CPTS_INTSTAT_RAW Register Field Descriptions

Bit	Field	Type	Reset	Description
0	TS_PEND_RAW	R/W	0h	TS_PEND_RAW int read (before enable). Writable when int_test = 1 A one in this bit indicates that there is one or more events in the event FIFO.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.7.6 CPTS_INTSTAT_MASKED Register (offset = h + (24h)) [reset = 0h]

TIME SYNC INTERRUPT STATUS MASKED REGISTER

CPTS_INTSTAT_MASKED is shown in Figure 13-115 and described in Table 13-124.

Figure 13-115. CPTS_INTSTAT_MASKED Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															TS_PEND
																															R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-124. CPTS_INTSTAT_MASKED Register Field Descriptions

Bit	Field	Type	Reset	Description
0	TS_PEND	R	0h	TS_PEND masked interrupt read (after enable).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.7.7 CPTS_CONTROL Register (offset = h + (4h)) [reset = 0h]

TIME SYNC CONTROL REGISTER

CPTS_CONTROL is shown in Figure 13-116 and described in Table 13-125.

Figure 13-116. CPTS_CONTROL Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved				HW4_TS_PUSH_EN	HW3_TS_PUSH_EN	HW2_TS_PUSH_EN	HW1_TS_PUSH_EN
				R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
Reserved						INT_TEST	CPTS_EN
						R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-125. CPTS_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
11	HW4_TS_PUSH_EN	R/W	0h	Hardware push 4 enable
10	HW3_TS_PUSH_EN	R/W	0h	Hardware push 3 enable
9	HW2_TS_PUSH_EN	R/W	0h	Hardware push 2 enable
8	HW1_TS_PUSH_EN	R/W	0h	Hardware push 1 enable
1	INT_TEST	R/W	0h	Interrupt Test - When set, this bit allows the raw interrupt to be written to facilitate interrupt test.
0	CPTS_EN	R/W	0h	Time Sync Enable - When disabled (cleared to zero), the RCLK domain is held in reset. 0 - Time Sync Disabled 1 - Time Sync Enabled

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.7.8 CPTS_INT_ENABLE Register (offset = h + (28h)) [reset = 0h]

TIME SYNC INTERRUPT ENABLE REGISTER

CPTS_INT_ENABLE is shown in Figure 13-117 and described in Table 13-126.

Figure 13-117. CPTS_INT_ENABLE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															TS_PEND_EN
																															R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-126. CPTS_INT_ENABLE Register Field Descriptions

Bit	Field	Type	Reset	Description
0	TS_PEND_EN	R/W	0h	TS_PEND masked interrupt enable.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.7.9 CPTS_EVENT_POP Register (offset = h + (30h)) [reset = 0h]

EVENT INTERRUPT POP REGISTER

CPTS_EVENT_POP is shown in Figure 13-118 and described in Table 13-127.

Figure 13-118. CPTS_EVENT_POP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															EVENT_POP
																															W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-127. CPTS_EVENT_POP Register Field Descriptions

Bit	Field	Type	Reset	Description
0	EVENT_POP	W	0h	Event Pop - When a logic high is written to this bit an event is popped off the event FIFO. The event FIFO pop occurs as part of the interrupt process after the event has been read in the Event_Low and Event_High registers. Popping an event discards the event and causes the next event, if any, to be moved to the top of the FIFO ready to be read by software on interrupt.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.7.10 CPTS_EVENT_LOW Register (offset = h + (34h)) [reset = 0h]

LOWER 32-BITS OF THE EVENT VALUE

CPTS_EVENT_LOW is shown in Figure 13-119 and described in Table 13-128.

Figure 13-119. CPTS_EVENT_LOW Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIME_STAMP																															
R-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-128. CPTS_EVENT_LOW Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TIME_STAMP	R	0h	Time Stamp - The timestamp is valid for transmit, receive, and time stamp push event types. The timestamp value is not valid for counter roll event types.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.7.11 CPTS_EVENT_HIGH Register (offset = h + (38h)) [reset = 0h]

UPPER 32-BITS OF THE EVENT VALUE

CPTS_EVENT_HIGH is shown in Figure 13-120 and described in Table 13-129.

Figure 13-120. CPTS_EVENT_HIGH Register

31	30	29	28	27	26	25	24
Reserved			PORT_NUMBER				
					R-0h		

23	22	21	20	19	18	17	16
EVENT_TYPE				MESSAGE_TYPE			
R-0h				R-0h			

15	14	13	12	11	10	9	8
SEQUENCE_ID							
R-0h							

7	6	5	4	3	2	1	0
SEQUENCE_ID							
R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-129. CPTS_EVENT_HIGH Register Field Descriptions

Bit	Field	Type	Reset	Description
28-24	PORT_NUMBER	R	0h	Port Number - indicates the port number of an ethernet event.
23-20	EVENT_TYPE	R	0h	Time Sync Event Type 0000 - Time Stamp Push Event 0001 - Time Stamp Rollover Event 0010 - Time Stamp Half Rollover Event 0011 - Hardware Time Stamp Push Event 0100 - Ethernet Receive Event 0101 - Ethernet Transmit Event
19-16	MESSAGE_TYPE	R	0h	Message type - The message type value that was contained in an ethernet transmit or receive time sync packet. This field is valid only for ethernet transmit or receive events.
15-0	SEQUENCE_ID	R	0h	Sequence ID - The 16-bit sequence id is the value that was contained in an ethernet transmit or receivetime sync packet. This field is valid only

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.7.12 CPTS_RFTCLK_SEL Register (offset = h + (8h)) [reset = 0h]

REFERENCE CLOCK SELECT REGISTER

CPTS_RFTCLK_SEL is shown in Figure 13-121 and described in Table 13-130.

Figure 13-121. CPTS_RFTCLK_SEL Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																											RFTCLK_SEL				
																											R/W-0h				

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-130. CPTS_RFTCLK_SEL Register Field Descriptions

Bit	Field	Type	Reset	Description
4-0	RFTCLK_SEL	R/W	0h	Reference Clock Select - This signal is used to control an external multiplexor that selects one of up to 32 clocks for time sync reference (RFTCLK). This rftclk_sel value can be written only when the cpts_en bit is cleared to zero in the TS_Control register.

13.8 CPSW_PORT Registers

Table 13-131 lists the memory-mapped registers for the CPSW_PORT. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 13-131 should be considered as reserved locations and the register contents should not be modified.

Table 13-131. CPSW_PORT REGISTERS

Offset	Acronym	Register Description	Section
0h	P0_CONTROL	CPSW PORT 0 CONTROL REGISTER	Section 13.8.1
Ch	P0_BLK_CNT	CPSW PORT 0 FIFO BLOCK USAGE COUNT (READ ONLY)	Section 13.8.2
10h	P0_TX_IN_CTL	CPSW PORT 0 TRANSMIT FIFO CONTROL	Section 13.8.3
14h	P0_PORT_VLAN	CPSW PORT 0 VLAN REGISTER	Section 13.8.4
18h	P0_TX_PRI_MAP	CPSW PORT 0 TX HEADER PRI TO SWITCH PRI MAPPING REGISTER	Section 13.8.5
100h	P1_CONTROL	CPSW PORT 1 CONTROL REGISTER	Section 13.8.6
108h	P1_MAX_BLKS	CPSW PORT 1 MAXIMUM FIFO BLOCKS REGISTER	Section 13.8.7
10Ch	P1_BLK_CNT	CPSW PORT 1 FIFO BLOCK USAGE COUNT (READ ONLY)	Section 13.8.8
110h	P1_TX_IN_CTL	CPSW PORT 1 TRANSMIT FIFO CONTROL	Section 13.8.9
114h	P1_PORT_VLAN	CPSW PORT 1 VLAN REGISTER	Section 13.8.10
1Ch	P0_CPDMA_TX_PRI_MAP	CPSW CPDMA TX (PORT 0 RX) PKT PRIORITY TO HEADER PRIORITY	Section 13.8.11
118h	P1_TX_PRI_MAP	CPSW PORT 1 TX HEADER PRIORITY TO SWITCH PRI MAPPING REGISTER	Section 13.8.12
11Ch	P1_TS_SEQ_MTYPE	CPSW PORT 1 TIME SYNC SEQUENCE ID OFFSET AND MSG TYPE.	Section 13.8.13
120h	P1_SA_LO	CPSW CPGMAC_SL1 SOURCE ADDRESS LOW REGISTER	Section 13.8.14
124h	P1_SA_HI	CPSW CPGMAC_SL1 SOURCE ADDRESS HIGH REGISTER	Section 13.8.15
128h	P1_SEND_PERCENT	CPSW PORT 1 TRANSMIT QUEUE SEND PERCENTAGES	Section 13.8.16
130h	P1_RX_DSCP_PRI_MAP0	CPSW PORT 1 RX DSCP PRIORITY TO RX PACKET MAPPING REG 0	Section 13.8.17

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 13-131. CPSW_PORT REGISTERS (continued)

Offset	Acronym	Register Description	Section
134h	P1_RX_DSCP_PRI_MAP1	CPSW PORT 1 RX DSCP PRIORITY TO RX PACKET MAPPING REG 1	Section 13.8.18
138h	P1_RX_DSCP_PRI_MAP2	CPSW PORT 1 RX DSCP PRIORITY TO RX PACKET MAPPING REG 2	Section 13.8.19
13Ch	P1_RX_DSCP_PRI_MAP3	CPSW PORT 1 RX DSCP PRIORITY TO RX PACKET MAPPING REG 3	Section 13.8.20
20h	P0_CPDMA_RX_CH_MAP	CPSW CPDMA RX (PORT 0 TX) SWITCH PRIORITY TO DMA CHANNEL	Section 13.8.21
140h	P1_RX_DSCP_PRI_MAP4	CPSW PORT 1 RX DSCP PRIORITY TO RX PACKET MAPPING REG 4	Section 13.8.22
144h	P1_RX_DSCP_PRI_MAP5	CPSW PORT 1 RX DSCP PRIORITY TO RX PACKET MAPPING REG 5	Section 13.8.23
148h	P1_RX_DSCP_PRI_MAP6	CPSW PORT 1 RX DSCP PRIORITY TO RX PACKET MAPPING REG 6	Section 13.8.24
14Ch	P1_RX_DSCP_PRI_MAP7	CPSW PORT 1 RX DSCP PRIORITY TO RX PACKET MAPPING REG 7	Section 13.8.25
30h	P0_RX_DSCP_PRI_MAP0	CPSW PORT 0 RX DSCP PRIORITY TO RX PACKET MAPPING REG 0	Section 13.8.26
200h	P2_CONTROL	CPSW_3GF PORT 2 CONTROL REGISTER	Section 13.8.27
34h	P0_RX_DSCP_PRI_MAP1	CPSW PORT 0 RX DSCP PRIORITY TO RX PACKET MAPPING REG 1	Section 13.8.28
208h	P2_MAX_BLKS	CPSW PORT 2 MAXIMUM FIFO BLOCKS REGISTER	Section 13.8.29
20Ch	P2_BLK_CNT	CPSW PORT 2 FIFO BLOCK USAGE COUNT (READ ONLY)	Section 13.8.30
210h	P2_TX_IN_CTL	CPSW PORT 2 TRANSMIT FIFO CONTROL	Section 13.8.31
214h	P2_PORT_VLAN	CPSW PORT 2 VLAN REGISTER	Section 13.8.32
218h	P2_TX_PRI_MAP	CPSW PORT 2 TX HEADER PRIORITY TO SWITCH PRI MAPPING REGISTER	Section 13.8.33
21Ch	P2_TS_SEQ_MTYPE	CPSW_3GF PORT 2 TIME SYNC SEQUENCE ID OFFSET AND MSG TYPE.	Section 13.8.34
220h	P2_SA_LO	CPSW CPGMAC_SL2 SOURCE ADDRESS LOW REGISTER	Section 13.8.35
224h	P2_SA_HI	CPSW CPGMAC_SL2 SOURCE ADDRESS HIGH REGISTER	Section 13.8.36
228h	P2_SEND_PERCENT	CPSW PORT 2 TRANSMIT QUEUE SEND PERCENTAGES	Section 13.8.37
38h	P0_RX_DSCP_PRI_MAP2	CPSW PORT 0 RX DSCP PRIORITY TO RX PACKET MAPPING REG 2	Section 13.8.38
230h	P2_RX_DSCP_PRI_MAP0	CPSW PORT 2 RX DSCP PRIORITY TO RX PACKET MAPPING REG 0	Section 13.8.39
234h	P2_RX_DSCP_PRI_MAP1	CPSW PORT 2 RX DSCP PRIORITY TO RX PACKET MAPPING REG 1	Section 13.8.40
238h	P2_RX_DSCP_PRI_MAP2	CPSW PORT 2 RX DSCP PRIORITY TO RX PACKET MAPPING REG 2	Section 13.8.41
23Ch	P2_RX_DSCP_PRI_MAP3	CPSW PORT 2 RX DSCP PRIORITY TO RX PACKET MAPPING REG 3	Section 13.8.42
240h	P2_RX_DSCP_PRI_MAP4	CPSW PORT 2 RX DSCP PRIORITY TO RX PACKET MAPPING REG 4	Section 13.8.43
244h	P2_RX_DSCP_PRI_MAP5	CPSW PORT 2 RX DSCP PRIORITY TO RX PACKET MAPPING REG 5	Section 13.8.44
248h	P2_RX_DSCP_PRI_MAP6	CPSW PORT 2 RX DSCP PRIORITY TO RX PACKET MAPPING REG 6	Section 13.8.45
24Ch	P2_RX_DSCP_PRI_MAP7	CPSW PORT 2 RX DSCP PRIORITY TO RX PACKET MAPPING REG 7	Section 13.8.46
3Ch	P0_RX_DSCP_PRI_MAP3	CPSW PORT 0 RX DSCP PRIORITY TO RX PACKET MAPPING REG 3	Section 13.8.47

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 13-131. CPSW_PORT REGISTERS (continued)

Offset	Acronym	Register Description	Section
40h	P0_RX_DSCP_PRI_MAP4	CPSW PORT 0 RX DSCP PRIORITY TO RX PACKET MAPPING REG 4	Section 13.8.48
44h	P0_RX_DSCP_PRI_MAP5	CPSW PORT 0 RX DSCP PRIORITY TO RX PACKET MAPPING REG 5	Section 13.8.49
48h	P0_RX_DSCP_PRI_MAP6	CPSW PORT 0 RX DSCP PRIORITY TO RX PACKET MAPPING REG 6	Section 13.8.50
4Ch	P0_RX_DSCP_PRI_MAP7	CPSW PORT 0 RX DSCP PRIORITY TO RX PACKET MAPPING REG 7	Section 13.8.51
8h	P0_MAX_BLKS	CPSW PORT 0 MAXIMUM FIFO BLOCKS REGISTER	Section 13.8.52

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.1 P0_CONTROL Register (offset = 0h) [reset = 0h]

CPSW PORT 0 CONTROL REGISTER

P0_CONTROL is shown in Figure 13-122 and described in Table 13-132.

Figure 13-122. P0_CONTROL Register

31	30	29	28	27	26	25	24
Reserved	P0_DLR_CPDMA_CH			Reserved			P0_PASS_PRI_TAGGED
	R/W-0h						R/W-0h

23	22	21	20	19	18	17	16
Reserved		P0_VLAN_LTYPE2_EN	P0_VLAN_LTYPE1_EN	Reserved			P0_DSCP_PRI_EN
		R/W-0h	R/W-0h				R/W-0h

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-132. P0_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
30-28	P0_DLR_CPDMA_CH	R/W	0h	Port 0 DLR CPDMA Channel This field indicates the CPDMA channel that DLR packets will be received on.
24	P0_PASS_PRI_TAGGED	R/W	0h	Port 0 Pass Priority Tagged 0 - Priority tagged packets have the zero VID replaced with the input port P0_PORT_VLAN[11:0] 1 - Priority tagged packets are processed unchanged.
21	P0_VLAN_LTYPE2_EN	R/W	0h	Port 0 VLAN LTYPE 2 enable 0 - disabled 1 - enabled
20	P0_VLAN_LTYPE1_EN	R/W	0h	Port 0 VLAN LTYPE 1 enable 0 - disabled 1 - enabled
16	P0_DSCP_PRI_EN	R/W	0h	Port 0 DSCP Priority Enable 0 - DSCP priority disabled 1 - DSCP priority enabled. All non-tagged IPV4 packets have their received packet priority determined by mapping the 6 TOS bits through the port DSCP priority mapping registers.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.2 P0_BLK_CNT Register (offset = Ch) [reset = 41h]

CPSW PORT 0 FIFO BLOCK USAGE COUNT (READ ONLY)

P0_BLK_CNT is shown in Figure 13-123 and described in Table 13-133.

Figure 13-123. P0_BLK_CNT Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							P0_TX_BLK_CNT
							R-4h

7	6	5	4	3	2	1	0
P0_TX_BLK_CNT				P0_RX_BLK_CNT			
R-4h				R-1h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-133. P0_BLK_CNT Register Field Descriptions

Bit	Field	Type	Reset	Description
8-4	P0_TX_BLK_CNT	R	4h	Port 0 Transmit Block Count Usage - This value is the number of blocks allocated to the FIFO logical transmit queues.
3-0	P0_RX_BLK_CNT	R	1h	Port 0 Receive Block Count Usage - This value is the number of blocks allocated to the FIFO logical receive queues.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.3 P0_TX_IN_CTL Register (offset = 10h) [reset = 40C0h]

CPSW PORT 0 TRANSMIT FIFO CONTROL

P0_TX_IN_CTL is shown in Figure 13-124 and described in Table 13-134.

Figure 13-124. P0_TX_IN_CTL Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
TX_RATE_EN				Reserved		TX_IN_SEL	
R/W-0h						R/W-0h	

15	14	13	12	11	10	9	8
TX_BLKS_REM				Reserved		TX_PRI_WDS	
R/W-4h						R/W-C0h	

7	6	5	4	3	2	1	0
TX_PRI_WDS							
R/W-C0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-134. P0_TX_IN_CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
23-20	TX_RATE_EN	R/W	0h	Transmit FIFO Input Rate Enable
17-16	TX_IN_SEL	R/W	0h	Transmit FIFO Input Queue Type Select 00 - Normal priority mode 01 - Dual MAC mode 10 - Rate Limit mode 11 - reserved Note that Dual MAC mode is not compatible with escalation or shaping because dual mac mode forces round robin priority on FIFO egress.
15-12	TX_BLKS_REM	R/W	4h	Transmit FIFO Input Blocks to subtract in dual mac mode
9-0	TX_PRI_WDS	R/W	C0h	Transmit FIFO Words in queue

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.4 P0_PORT_VLAN Register (offset = 14h) [reset = 0h]

CPSW PORT 0 VLAN REGISTER

P0_PORT_VLAN is shown in Figure 13-125 and described in Table 13-135.

Figure 13-125. P0_PORT_VLAN Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
PORT_PRI			PORT_CFI	PORT_VID			
R/W-0h			R/W-0h	R/W-0h			

7	6	5	4	3	2	1	0
PORT_VID							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-135. P0_PORT_VLAN Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	PORT_PRI	R/W	0h	Port VLAN Priority (7 is highest priority)
12	PORT_CFI	R/W	0h	Port CFI bit
11-0	PORT_VID	R/W	0h	Port VLAN ID

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.5 P0_TX_PRI_MAP Register (offset = 18h) [reset = 33221001h]

CPSW PORT 0 TX HEADER PRI TO SWITCH PRI MAPPING REGISTER

P0_TX_PRI_MAP is shown in Figure 13-126 and described in Table 13-136.

Figure 13-126. P0_TX_PRI_MAP Register

31	30	29	28	27	26	25	24
Reserved		PRI7		Reserved		PRI6	
		R/W-3h				R/W-3h	

23	22	21	20	19	18	17	16
Reserved		PRI5		Reserved		PRI4	
		R/W-2h				R/W-2h	

15	14	13	12	11	10	9	8
Reserved		PRI3		Reserved		PRI2	
		R/W-1h				R/W-0h	

7	6	5	4	3	2	1	0
Reserved		PRI1		Reserved		PRI0	
		R/W-0h				R/W-1h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-136. P0_TX_PRI_MAP Register Field Descriptions

Bit	Field	Type	Reset	Description
29-28	PRI7	R/W	3h	Priority 7 - A packet header priority of 0x7 is given this switch queue pri.
25-24	PRI6	R/W	3h	Priority 6 - A packet header priority of 0x6 is given this switch queue pri.
21-20	PRI5	R/W	2h	Priority 5 - A packet header priority of 0x5 is given this switch queue pri.
17-16	PRI4	R/W	2h	Priority 4 - A packet header priority of 0x4 is given this switch queue pri.
13-12	PRI3	R/W	1h	Priority 3 - A packet header priority of 0x3 is given this switch queue pri.
9-8	PRI2	R/W	0h	Priority 2 - A packet header priority of 0x2 is given this switch queue pri.
5-4	PRI1	R/W	0h	Priority 1 - A packet header priority of 0x1 is given this switch queue pri.
1-0	PRI0	R/W	1h	Priority 0 - A packet header priority of 0x0 is given this switch queue pri.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.6 P1_CONTROL Register (offset = 100h) [reset = 0h]

CPSW PORT 1 CONTROL REGISTER

P1_CONTROL is shown in Figure 13-127 and described in Table 13-137.

Figure 13-127. P1_CONTROL Register

31	30	29	28	27	26	25	24
Reserved							P1_PASS_PRI_TAGGED
							R/W-0h

23	22	21	20	19	18	17	16
Reserved		P1_VLAN_LTYPE2_EN	P1_VLAN_LTYPE1_EN	Reserved			P1_DSCP_PRI_EN
		R/W-0h	R/W-0h				R/W-0h

15	14	13	12	11	10	9	8
Reserved	P1_TS_320	P1_TS_319	P1_TS_132	P1_TS_131	P1_TS_130	P1_TS_129	Reserved
	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	

7	6	5	4	3	2	1	0
Reserved			P1_TS_ANNEX_D_EN	P1_TS_LTYPE2_EN	P1_TS_LTYPE1_EN	P1_TS_TX_EN	P1_TS_RX_EN
			R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-137. P1_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
24	P1_PASS_PRI_TAGGED	R/W	0h	Port 1 Pass Priority Tagged 0 - Priority tagged packets have the zero VID replaced with the input port P1_PORT_VLAN[11:0] 1 - Priority tagged packets are processed unchanged.
21	P1_VLAN_LTYPE2_EN	R/W	0h	Port 1 VLAN LTYPE 2 enable 0 - disabled 1 - VLAN LTYPE2 enabled on transmit and receive
20	P1_VLAN_LTYPE1_EN	R/W	0h	Port 1 VLAN LTYPE 1 enable 0 - disabled 1 - VLAN LTYPE1 enabled on transmit and receive
16	P1_DSCP_PRI_EN	R/W	0h	Port 1 DSCP Priority Enable 0 - DSCP priority disabled 1 - DSCP priority enabled. All non-tagged IPV4 packets have their received packet priority determined by mapping the 6 TOS bits through the port DSCP priority mapping registers.
14	P1_TS_320	R/W	0h	Port 1 Time Sync Destination Port Number 320 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination port number 320 (decimal) is enabled.
13	P1_TS_319	R/W	0h	Port 1 Time Sync Destination Port Number 319 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination port number 319 (decimal) is enabled.
12	P1_TS_132	R/W	0h	Port 1 Time Sync Destination IP Address 132 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination IP address number 132 (decimal) is enabled.
11	P1_TS_131	R/W	0h	Port 1 Time Sync Destination IP Address 131 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination IP address number 131 (decimal) is enabled.
10	P1_TS_130	R/W	0h	Port 1 Time Sync Destination IP Address 130 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination IP address number 130 (decimal) is enabled.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 13-137. P1_CONTROL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
9	P1_TS_129	R/W	0h	Port 1 Time Sync Destination IP Address 129 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination IP address number 129 (decimal) is enabled.
4	P1_TS_ANNEX_D_EN	R/W	0h	Port 1 Time Sync Annex D enable 0 - Annex D disabled 1 - Annex D enabled
3	P1_TS_LTYPE2_EN	R	0h	Port 1 Time Sync LTYPE 2 enable 0 - disabled 1 - enabled
2	P1_TS_LTYPE1_EN	R/W	0h	Port 1 Time Sync LTYPE 1 enable 0 - disabled 1 - enabled
1	P1_TS_TX_EN	R/W	0h	Port 1 Time Sync Transmit Enable 0 - disabled 1 - enabled
0	P1_TS_RX_EN	R/W	0h	Port 1 Time Sync Receive Enable 0 - Port 1 Receive Time Sync disabled 1 - Port 1 Receive Time Sync enabled

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.7 P1_MAX_BLKS Register (offset = 108h) [reset = 113h]

CPSW PORT 1 MAXIMUM FIFO BLOCKS REGISTER

P1_MAX_BLKS is shown in Figure 13-128 and described in Table 13-138.

Figure 13-128. P1_MAX_BLKS Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							P1_TX_MAX_BLKS
							R/W-11h

7	6	5	4	3	2	1	0
P1_TX_MAX_BLKS				P1_RX_MAX_BLKS			
R/W-11h				R/W-3h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-138. P1_MAX_BLKS Register Field Descriptions

Bit	Field	Type	Reset	Description
8-4	P1_TX_MAX_BLKS	R/W	11h	Transmit FIFO Maximum Blocks - This value is the maximum number of 1k memory blocks that may be allocated to the FIFO's logical transmit priority queues. 0x11 is the recommended value of p1_tx_max_blks unless the port is in fullduplex flow control mode. In flow control mode, the p1_rx_max_blks will need to increase in order to accept the required run out in fullduplex mode. This value will need to decrease by the amount of increase in p1_rx_max_blks. 0xe is the minimum value tx max blks.
3-0	P1_RX_MAX_BLKS	R/W	3h	Receive FIFO Maximum Blocks - This value is the maximum number of 1k memory blocks that may be allocated to the FIFO's logical receive queue. This value must be greater than or equal to 0x3. It should be increased In fullduplex flow control mode to 0x5 or 0x6 depending on the required runout space. The p1_tx_max_blks value must be decreased by the amount of increase in p1_rx_max_blks. 0x3 is the minimum value rx max blks and 0x6 is the maximum value.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.8 P1_BLK_CNT Register (offset = 10Ch) [reset = 41h]

CPSW PORT 1 FIFO BLOCK USAGE COUNT (READ ONLY)

P1_BLK_CNT is shown in Figure 13-129 and described in Table 13-139.

Figure 13-129. P1_BLK_CNT Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							P1_TX_BLK_CNT
							R-4h

7	6	5	4	3	2	1	0
P1_TX_BLK_CNT				P1_RX_BLK_CNT			
R-4h				R-1h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-139. P1_BLK_CNT Register Field Descriptions

Bit	Field	Type	Reset	Description
8-4	P1_TX_BLK_CNT	R	4h	Port 1 Transmit Block Count Usage - This value is the number of blocks allocated to the FIFO logical transmit queues.
3-0	P1_RX_BLK_CNT	R	1h	Port 1 Receive Block Count Usage - This value is the number of blocks allocated to the FIFO logical receive queues.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.9 P1_TX_IN_CTL Register (offset = 110h) [reset = 80040C0h]

CPSW PORT 1 TRANSMIT FIFO CONTROL

P1_TX_IN_CTL is shown in Figure 13-130 and described in Table 13-140.

Figure 13-130. P1_TX_IN_CTL Register

31	30	29	28	27	26	25	24
Reserved				HOST_BLKS_REM			
				R/W-8h			

23	22	21	20	19	18	17	16
TX_RATE_EN				Reserved		TX_IN_SEL	
R/W-0h						R/W-0h	

15	14	13	12	11	10	9	8
TX_BLKS_REM				Reserved		TX_PRI_WDS	
R/W-4h						R/W-C0h	

7	6	5	4	3	2	1	0
TX_PRI_WDS							
R/W-C0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-140. P1_TX_IN_CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
27-24	HOST_BLKS_REM	R/W	8h	Transmit FIFO Blocks that must be free before a non rate-limited CPDMA channel can begin sending a packet to the FIFO.
23-20	TX_RATE_EN	R/W	0h	Transmit FIFO Input Rate Enable
17-16	TX_IN_SEL	R/W	0h	Transmit FIFO Input Queue Type Select 00 - Normal priority mode 01 - reserved 10 - Rate Limit mode 11 - reserved
15-12	TX_BLKS_REM	R/W	4h	Transmit FIFO Input Blocks to subtract in dual mac mode and blocks to subtract on non rate-limited traffic in rate-limit mode.
9-0	TX_PRI_WDS	R/W	C0h	Transmit FIFO Words in queue

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.10 P1_PORT_VLAN Register (offset = 114h) [reset = 0h]

CPSW PORT 1 VLAN REGISTER

P1_PORT_VLAN is shown in Figure 13-131 and described in Table 13-141.

Figure 13-131. P1_PORT_VLAN Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
PORT_PRI			PORT_CFI	PORT_VID			
R/W-0h			R/W-0h	R/W-0h			

7	6	5	4	3	2	1	0
PORT_VID							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-141. P1_PORT_VLAN Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	PORT_PRI	R/W	0h	Port VLAN Priority (7 is highest priority)
12	PORT_CFI	R/W	0h	Port CFI bit
11-0	PORT_VID	R/W	0h	Port VLAN ID

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.11 P0_CPDMA_TX_PRI_MAP Register (offset = 1Ch) [reset = 76543210h]

CPSW CPDMA TX (PORT 0 RX) PKT PRIORITY TO HEADER PRIORITY

P0_CPDMA_TX_PRI_MAP is shown in Figure 13-132 and described in Table 13-142.

Figure 13-132. P0_CPDMA_TX_PRI_MAP Register

31	30	29	28	27	26	25	24
Reserved	PRI7			Reserved	PRI6		
		R/W-7h				R/W-6h	

23	22	21	20	19	18	17	16
Reserved	PRI5			Reserved	PRI4		
		R/W-5h				R/W-4h	

15	14	13	12	11	10	9	8
Reserved	PRI3			Reserved	PRI2		
		R/W-3h				R/W-2h	

7	6	5	4	3	2	1	0
Reserved	PRI1			Reserved	PRI0		
		R/W-1h				R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-142. P0_CPDMA_TX_PRI_MAP Register Field Descriptions

Bit	Field	Type	Reset	Description
30-28	PRI7	R/W	7h	Priority 7 - A packet pri of 0x7 is mapped (changed) to this header packet pri.
26-24	PRI6	R/W	6h	Priority 6 - A packet pri of 0x6 is mapped (changed) to this header packet pri.
22-20	PRI5	R/W	5h	Priority 5 - A packet pri of 0x5 is mapped (changed) to this header packet pri.
18-16	PRI4	R/W	4h	Priority 4 - A packet pri of 0x4 is mapped (changed) to this header packet pri.
14-12	PRI3	R/W	3h	Priority 3 - A packet pri of 0x3 is mapped (changed) to this header packet pri.
10-8	PRI2	R/W	2h	Priority 2 - A packet pri of 0x2 is mapped (changed) to this header packet pri.
6-4	PRI1	R/W	1h	Priority 1 - A packet pri of 0x1 is mapped (changed) to this header packet pri.
2-0	PRI0	R/W	0h	Priority 0 - A packet pri of 0x0 is mapped (changed) to this header packet pri.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.12 P1_TX_PRI_MAP Register (offset = 118h) [reset = 33221001h]

CPSW PORT 1 TX HEADER PRIORITY TO SWITCH PRI MAPPING REGISTER

P1_TX_PRI_MAP is shown in Figure 13-133 and described in Table 13-143.

Figure 13-133. P1_TX_PRI_MAP Register

31	30	29	28	27	26	25	24
Reserved		PRI7		Reserved		PRI6	
		R/W-3h				R/W-3h	

23	22	21	20	19	18	17	16
Reserved		PRI5		Reserved		PRI4	
		R/W-2h				R/W-2h	

15	14	13	12	11	10	9	8
Reserved		PRI3		Reserved		PRI2	
		R/W-1h				R/W-0h	

7	6	5	4	3	2	1	0
Reserved		PRI1		Reserved		PRI0	
		R/W-0h				R/W-1h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-143. P1_TX_PRI_MAP Register Field Descriptions

Bit	Field	Type	Reset	Description
29-28	PRI7	R/W	3h	Priority 7 - A packet header priority of 0x7 is given this switch queue pri.
25-24	PRI6	R/W	3h	Priority 6 - A packet header priority of 0x6 is given this switch queue pri.
21-20	PRI5	R/W	2h	Priority 5 - A packet header priority of 0x5 is given this switch queue pri.
17-16	PRI4	R/W	2h	Priority 4 - A packet header priority of 0x4 is given this switch queue pri.
13-12	PRI3	R/W	1h	Priority 3 - A packet header priority of 0x3 is given this switch queue pri.
9-8	PRI2	R/W	0h	Priority 2 - A packet header priority of 0x2 is given this switch queue pri.
5-4	PRI1	R/W	0h	Priority 1 - A packet header priority of 0x1 is given this switch queue pri.
1-0	PRI0	R/W	1h	Priority 0 - A packet header priority of 0x0 is given this switch queue pri.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.13 P1_TS_SEQ_MTYPE Register (offset = 11Ch) [reset = 1E0000h]

CPSW PORT 1 TIME SYNC SEQUENCE ID OFFSET AND MSG TYPE.

P1_TS_SEQ_MTYPE is shown in Figure 13-134 and described in Table 13-144.

Figure 13-134. P1_TS_SEQ_MTYPE Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved		P1_TS_SEQ_ID_OFFSET					
		R/W-1Eh					

15	14	13	12	11	10	9	8
P1_TS_MSG_TYPE_EN							
R/W-0h							

7	6	5	4	3	2	1	0
P1_TS_MSG_TYPE_EN							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-144. P1_TS_SEQ_MTYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
21-16	P1_TS_SEQ_ID_OFFSET	R/W	1Eh	Port 1 Time Sync Sequence ID Offset This is the number of octets that the sequence ID is offset in the tx and rx time sync message header. The minimum value is 6.
15-0	P1_TS_MSG_TYPE_EN	R/W	0h	Port 1 Time Sync Message Type Enable - Each bit in this field enables the corresponding message type in receive and transmit time sync messages (Bit 0 enables message type 0 etc. .

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.14 P1_SA_LO Register (offset = 120h) [reset = 0h]

CPSW CPGMAC_SL1 SOURCE ADDRESS LOW REGISTER

P1_SA_LO is shown in Figure 13-135 and described in Table 13-145.

Figure 13-135. P1_SA_LO Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
MACSRCADDR_7_0							
R/W-0h							

7	6	5	4	3	2	1	0
MACSRCADDR_15_8							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-145. P1_SA_LO Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	MACSRCADDR_7_0	R/W	0h	Source Address Lower 8 bits (byte 0)
7-0	MACSRCADDR_15_8	R/W	0h	Source Address bits 15:8 (byte 1)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.15 P1_SA_HI Register (offset = 124h) [reset = 0h]

CPSW CPGMAC_SL1 SOURCE ADDRESS HIGH REGISTER

P1_SA_HI is shown in Figure 13-136 and described in Table 13-146.

Figure 13-136. P1_SA_HI Register

31	30	29	28	27	26	25	24
MACSRCADDR_23_16							
R/W-0h							

23	22	21	20	19	18	17	16
MACSRCADDR_31_24							
R/W-0h							

15	14	13	12	11	10	9	8
MACSRCADDR_39_32							
R/W-0h							

7	6	5	4	3	2	1	0
MACSRCADDR_47_40							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-146. P1_SA_HI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	MACSRCADDR_23_16	R/W	0h	Source Address bits 23:16 (byte 2)
23-16	MACSRCADDR_31_24	R/W	0h	Source Address bits 31:24 (byte 3)
15-8	MACSRCADDR_39_32	R/W	0h	Source Address bits 39:32 (byte 4)
7-0	MACSRCADDR_47_40	R/W	0h	Source Address bits 47:40 (byte 5)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.16 P1_SEND_PERCENT Register (offset = 128h) [reset = 0h]

CPSW PORT 1 TRANSMIT QUEUE SEND PERCENTAGES

P1_SEND_PERCENT is shown in Figure 13-137 and described in Table 13-147.

Figure 13-137. P1_SEND_PERCENT Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved	PRI3_SEND_PERCENT						
	R/W-0h						

15	14	13	12	11	10	9	8
Reserved	PRI2_SEND_PERCENT						
	R/W-0h						

7	6	5	4	3	2	1	0
Reserved	PRI1_SEND_PERCENT						
	R/W-0h						

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-147. P1_SEND_PERCENT Register Field Descriptions

Bit	Field	Type	Reset	Description
22-16	PRI3_SEND_PERCENT	R/W	0h	Priority 3 Transmit Percentage - This percentage value is sent from FIFO priority 3 (maximum) when the p1_pri3_shape_en is set (queue shaping enabled). This is the percentage of the wire that packets from priority 3 receive (which includes interpacket gap and preamble bytes). If shaping is enabled on this queue then this value must be between zero and 0d100 (not inclusive).
14-8	PRI2_SEND_PERCENT	R/W	0h	Priority 2 Transmit Percentage - This percentage value is sent from FIFO priority 2 (maximum) when the p1_pri2_shape_en is set (queue shaping enabled). This is the percentage of the wire that packets from priority 2 receive (which includes interpacket gap and preamble bytes). If shaping is enabled on this queue then this value must be between zero and 0d100 (not inclusive).
6-0	PRI1_SEND_PERCENT	R/W	0h	Priority 1 Transmit Percentage - This percentage value is sent from FIFO priority 1 (maximum) when the p1_pri1_shape_en is set (queue shaping enabled). This is the percentage of the wire that packets from priority 1 receive (which includes interpacket gap and preamble bytes). If shaping is enabled on this queue then this value must be between zero and 0d100 (not inclusive).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.17 P1_RX_DSCP_PRI_MAP0 Register (offset = 130h) [reset = 0h]

CPSW PORT 1 RX DSCP PRIORITY TO RX PACKET MAPPING REG 0

P1_RX_DSCP_PRI_MAP0 is shown in Figure 13-138 and described in Table 13-148.

Figure 13-138. P1_RX_DSCP_PRI_MAP0 Register

31	30	29	28	27	26	25	24
Reserved	PRI7			Reserved	PRI6		
		R/W-0h				R/W-0h	

23	22	21	20	19	18	17	16
Reserved	PRI5			Reserved	PRI4		
		R/W-0h				R/W-0h	

15	14	13	12	11	10	9	8
Reserved	PRI3			Reserved	PRI2		
		R/W-0h				R/W-0h	

7	6	5	4	3	2	1	0
Reserved	PRI1			Reserved	PRI0		
		R/W-0h				R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-148. P1_RX_DSCP_PRI_MAP0 Register Field Descriptions

Bit	Field	Type	Reset	Description
30-28	PRI7	R/W	0h	Priority 7 - A packet TOS of 0d7 is mapped to this received packet priority.
26-24	PRI6	R/W	0h	Priority 6 - A packet TOS of 0d6 is mapped to this received packet priority.
22-20	PRI5	R/W	0h	Priority 5 - A packet TOS of 0d5 is mapped to this received packet priority.
18-16	PRI4	R/W	0h	Priority 4 - A packet TOS of 0d4 is mapped to this received packet priority.
14-12	PRI3	R/W	0h	Priority 3 - A packet TOS of 0d3 is mapped to this received packet priority.
10-8	PRI2	R/W	0h	Priority 2 - A packet TOS of 0d2 is mapped to this received packet priority.
6-4	PRI1	R/W	0h	Priority 1 - A packet TOS of 0d1 is mapped to this received packet priority.
2-0	PRI0	R/W	0h	Priority 0 - A packet TOS of 0d0 is mapped to this received packet priority.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.18 P1_RX_DSCP_PRI_MAP1 Register (offset = 134h) [reset = 0h]

CPSW PORT 1 RX DSCP PRIORITY TO RX PACKET MAPPING REG 1

P1_RX_DSCP_PRI_MAP1 is shown in Figure 13-139 and described in Table 13-149.

Figure 13-139. P1_RX_DSCP_PRI_MAP1 Register

31	30	29	28	27	26	25	24
Reserved	PRI15			Reserved	PRI14		
		R/W-0h				R/W-0h	

23	22	21	20	19	18	17	16
Reserved	PRI13			Reserved	PRI12		
		R/W-0h				R/W-0h	

15	14	13	12	11	10	9	8
Reserved	PRI11			Reserved	PRI10		
		R/W-0h				R/W-0h	

7	6	5	4	3	2	1	0
Reserved	PRI9			Reserved	PRI8		
		R/W-0h				R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-149. P1_RX_DSCP_PRI_MAP1 Register Field Descriptions

Bit	Field	Type	Reset	Description
30-28	PRI15	R/W	0h	Priority 15 - A packet TOS of 0d15 is mapped to this received packet priority.
26-24	PRI14	R/W	0h	Priority 14 - A packet TOS of 0d14 is mapped to this received packet priority.
22-20	PRI13	R/W	0h	Priority 13 - A packet TOS of 0d13 is mapped to this received packet priority.
18-16	PRI12	R/W	0h	Priority 12 - A packet TOS of 0d12 is mapped to this received packet priority.
14-12	PRI11	R/W	0h	Priority 11 - A packet TOS of 0d11 is mapped to this received packet priority.
10-8	PRI10	R/W	0h	Priority 10 - A packet TOS of 0d10 is mapped to this received packet priority.
6-4	PRI9	R/W	0h	Priority 9 - A packet TOS of 0d9 is mapped to this received packet priority.
2-0	PRI8	R/W	0h	Priority 8 - A packet TOS of 0d8 is mapped to this received packet priority.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.19 P1_RX_DSCP_PRI_MAP2 Register (offset = 138h) [reset = 0h]

CPSW PORT 1 RX DSCP PRIORITY TO RX PACKET MAPPING REG 2

P1_RX_DSCP_PRI_MAP2 is shown in Figure 13-140 and described in Table 13-150.

Figure 13-140. P1_RX_DSCP_PRI_MAP2 Register

31	30	29	28	27	26	25	24
Reserved	PRI23			Reserved	PRI22		
		R/W-0h				R/W-0h	

23	22	21	20	19	18	17	16
Reserved	PRI21			Reserved	PRI20		
		R/W-0h				R/W-0h	

15	14	13	12	11	10	9	8
Reserved	PRI19			Reserved	PRI18		
		R/W-0h				R/W-0h	

7	6	5	4	3	2	1	0
Reserved	PRI17			Reserved	PRI16		
		R/W-0h				R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-150. P1_RX_DSCP_PRI_MAP2 Register Field Descriptions

Bit	Field	Type	Reset	Description
30-28	PRI23	R/W	0h	Priority 23 - A packet TOS of 0d23 is mapped to this received packet priority.
26-24	PRI22	R/W	0h	Priority 22 - A packet TOS of 0d22 is mapped to this received packet priority.
22-20	PRI21	R/W	0h	Priority 21 - A packet TOS of 0d21 is mapped to this received packet priority.
18-16	PRI20	R/W	0h	Priority 20 - A packet TOS of 0d20 is mapped to this received packet priority.
14-12	PRI19	R/W	0h	Priority 19 - A packet TOS of 0d19 is mapped to this received packet priority.
10-8	PRI18	R/W	0h	Priority 18 - A packet TOS of 0d18 is mapped to this received packet priority.
6-4	PRI17	R/W	0h	Priority 17 - A packet TOS of 0d17 is mapped to this received packet priority.
2-0	PRI16	R/W	0h	Priority 16 - A packet TOS of 0d16 is mapped to this received packet priority.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.20 P1_RX_DSCP_PRI_MAP3 Register (offset = 13Ch) [reset = 0h]

CPSW PORT 1 RX DSCP PRIORITY TO RX PACKET MAPPING REG 3

P1_RX_DSCP_PRI_MAP3 is shown in Figure 13-141 and described in Table 13-151.

Figure 13-141. P1_RX_DSCP_PRI_MAP3 Register

31	30	29	28	27	26	25	24
Reserved	PRI31			Reserved	PRI30		
		R/W-0h				R/W-0h	

23	22	21	20	19	18	17	16
Reserved	PRI29			Reserved	PRI28		
		R/W-0h				R/W-0h	

15	14	13	12	11	10	9	8
Reserved	PRI27			Reserved	PRI26		
		R/W-0h				R/W-0h	

7	6	5	4	3	2	1	0
Reserved	PRI25			Reserved	PRI24		
		R/W-0h				R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-151. P1_RX_DSCP_PRI_MAP3 Register Field Descriptions

Bit	Field	Type	Reset	Description
30-28	PRI31	R/W	0h	Priority 31 - A packet TOS of 0d31 is mapped to this received packet priority.
26-24	PRI30	R/W	0h	Priority 30 - A packet TOS of 0d30 is mapped to this received packet priority.
22-20	PRI29	R/W	0h	Priority 29 - A packet TOS of 0d39 is mapped to this received packet priority.
18-16	PRI28	R/W	0h	Priority 28 - A packet TOS of 0d28 is mapped to this received packet priority.
14-12	PRI27	R/W	0h	Priority 27 - A packet TOS of 0d27 is mapped to this received packet priority.
10-8	PRI26	R/W	0h	Priority 26 - A packet TOS of 0d26 is mapped to this received packet priority.
6-4	PRI25	R/W	0h	Priority 25 - A packet TOS of 0d25 is mapped to this received packet priority.
2-0	PRI24	R/W	0h	Priority 24 - A packet TOS of 0d24 is mapped to this received packet priority.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.21 P0_CPDMA_RX_CH_MAP Register (offset = 20h) [reset = 0h]

CPSW CPDMA RX (PORT 0 TX) SWITCH PRIORITY TO DMA CHANNEL

P0_CPDMA_RX_CH_MAP is shown in Figure 13-142 and described in Table 13-152.

Figure 13-142. P0_CPDMA_RX_CH_MAP Register

31	30	29	28	27	26	25	24
Reserved	P2_PRI3			Reserved	P2_PRI2		
	R/W-0h				R/W-0h		

23	22	21	20	19	18	17	16
Reserved	P2_PRI1			Reserved	P2_PRI0		
	R/W-0h				R/W-0h		

15	14	13	12	11	10	9	8
Reserved	P1_PRI3			Reserved	P1_PRI2		
	R/W-0h				R/W-0h		

7	6	5	4	3	2	1	0
Reserved	P1_PRI1			Reserved	P1_PRI0		
	R/W-0h				R/W-0h		

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-152. P0_CPDMA_RX_CH_MAP Register Field Descriptions

Bit	Field	Type	Reset	Description
30-28	P2_PRI3	R/W	0h	Port 2 Priority 3 packets go to this CPDMA Rx Channel
26-24	P2_PRI2	R/W	0h	Port 2 Priority 2 packets go to this CPDMA Rx Channel
22-20	P2_PRI1	R/W	0h	Port 2 Priority 1 packets go to this CPDMA Rx Channel
18-16	P2_PRI0	R/W	0h	Port 2 Priority 0 packets go to this CPDMA Rx Channel
14-12	P1_PRI3	R/W	0h	Port 1 Priority 3 packets go to this CPDMA Rx Channel
10-8	P1_PRI2	R/W	0h	Port 1 Priority 2 packets go to this CPDMA Rx Channel
6-4	P1_PRI1	R/W	0h	Port 1 Priority 1 packets go to this CPDMA Rx Channel
2-0	P1_PRI0	R/W	0h	Port 1 Priority 0 packets go to this CPDMA Rx Channel

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.22 P1_RX_DSCP_PRI_MAP4 Register (offset = 140h) [reset = 0h]

CPSW PORT 1 RX DSCP PRIORITY TO RX PACKET MAPPING REG 4

P1_RX_DSCP_PRI_MAP4 is shown in Figure 13-143 and described in Table 13-153.

Figure 13-143. P1_RX_DSCP_PRI_MAP4 Register

31	30	29	28	27	26	25	24
Reserved	PRI39			Reserved	PRI38		
		R/W-0h				R/W-0h	

23	22	21	20	19	18	17	16
Reserved	PRI37			Reserved	PRI36		
		R/W-0h				R/W-0h	

15	14	13	12	11	10	9	8
Reserved	PRI35			Reserved	PRI34		
		R/W-0h				R/W-0h	

7	6	5	4	3	2	1	0
Reserved	PRI33			Reserved	PRI32		
		R/W-0h				R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-153. P1_RX_DSCP_PRI_MAP4 Register Field Descriptions

Bit	Field	Type	Reset	Description
30-28	PRI39	R/W	0h	Priority 39 - A packet TOS of 0d39 is mapped to this received packet priority.
26-24	PRI38	R/W	0h	Priority 38 - A packet TOS of 0d38 is mapped to this received packet priority.
22-20	PRI37	R/W	0h	Priority 37 - A packet TOS of 0d37 is mapped to this received packet priority.
18-16	PRI36	R/W	0h	Priority 36 - A packet TOS of 0d36 is mapped to this received packet priority.
14-12	PRI35	R/W	0h	Priority 35 - A packet TOS of 0d35 is mapped to this received packet priority.
10-8	PRI34	R/W	0h	Priority 34 - A packet TOS of 0d34 is mapped to this received packet priority.
6-4	PRI33	R/W	0h	Priority 33 - A packet TOS of 0d33 is mapped to this received packet priority.
2-0	PRI32	R/W	0h	Priority 32 - A packet TOS of 0d32 is mapped to this received packet priority.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.23 P1_RX_DSCP_PRI_MAP5 Register (offset = 144h) [reset = 0h]

CPSW PORT 1 RX DSCP PRIORITY TO RX PACKET MAPPING REG 5

P1_RX_DSCP_PRI_MAP5 is shown in Figure 13-144 and described in Table 13-154.

Figure 13-144. P1_RX_DSCP_PRI_MAP5 Register

31	30	29	28	27	26	25	24
Reserved	PRI47			Reserved	PRI46		
		R/W-0h				R/W-0h	

23	22	21	20	19	18	17	16
Reserved	PRI45			Reserved	PRI44		
		R/W-0h				R/W-0h	

15	14	13	12	11	10	9	8
Reserved	PRI43			Reserved	PRI42		
		R/W-0h				R/W-0h	

7	6	5	4	3	2	1	0
Reserved	PRI41			Reserved	PRI40		
		R/W-0h				R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-154. P1_RX_DSCP_PRI_MAP5 Register Field Descriptions

Bit	Field	Type	Reset	Description
30-28	PRI47	R/W	0h	Priority 47 - A packet TOS of 0d47 is mapped to this received packet priority.
26-24	PRI46	R/W	0h	Priority 46 - A packet TOS of 0d46 is mapped to this received packet priority.
22-20	PRI45	R/W	0h	Priority 45 - A packet TOS of 0d45 is mapped to this received packet priority.
18-16	PRI44	R/W	0h	Priority 44 - A packet TOS of 0d44 is mapped to this received packet priority.
14-12	PRI43	R/W	0h	Priority 43 - A packet TOS of 0d43 is mapped to this received packet priority.
10-8	PRI42	R/W	0h	Priority 42 - A packet TOS of 0d42 is mapped to this received packet priority.
6-4	PRI41	R/W	0h	Priority 41 - A packet TOS of 0d41 is mapped to this received packet priority.
2-0	PRI40	R/W	0h	Priority 40 - A packet TOS of 0d40 is mapped to this received packet priority.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.24 P1_RX_DSCP_PRI_MAP6 Register (offset = 148h) [reset = 0h]

CPSW PORT 1 RX DSCP PRIORITY TO RX PACKET MAPPING REG 6

P1_RX_DSCP_PRI_MAP6 is shown in Figure 13-145 and described in Table 13-155.

Figure 13-145. P1_RX_DSCP_PRI_MAP6 Register

31	30	29	28	27	26	25	24
Reserved	PRI55			Reserved	PRI54		
		R/W-0h				R/W-0h	

23	22	21	20	19	18	17	16
Reserved	PRI53			Reserved	PRI52		
		R/W-0h				R/W-0h	

15	14	13	12	11	10	9	8
Reserved	PRI51			Reserved	PRI50		
		R/W-0h				R/W-0h	

7	6	5	4	3	2	1	0
Reserved	PRI49			Reserved	PRI48		
		R/W-0h				R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-155. P1_RX_DSCP_PRI_MAP6 Register Field Descriptions

Bit	Field	Type	Reset	Description
30-28	PRI55	R/W	0h	Priority 55 - A packet TOS of 0d55 is mapped to this received packet priority.
26-24	PRI54	R/W	0h	Priority 54 - A packet TOS of 0d54 is mapped to this received packet priority.
22-20	PRI53	R/W	0h	Priority 53 - A packet TOS of 0d53 is mapped to this received packet priority.
18-16	PRI52	R/W	0h	Priority 52 - A packet TOS of 0d52 is mapped to this received packet priority.
14-12	PRI51	R/W	0h	Priority 51 - A packet TOS of 0d51 is mapped to this received packet priority.
10-8	PRI50	R/W	0h	Priority 50 - A packet TOS of 0d50 is mapped to this received packet priority.
6-4	PRI49	R/W	0h	Priority 49 - A packet TOS of 0d49 is mapped to this received packet priority.
2-0	PRI48	R/W	0h	Priority 48 - A packet TOS of 0d48 is mapped to this received packet priority.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.25 P1_RX_DSCP_PRI_MAP7 Register (offset = 14Ch) [reset = 0h]

CPSW PORT 1 RX DSCP PRIORITY TO RX PACKET MAPPING REG 7

P1_RX_DSCP_PRI_MAP7 is shown in Figure 13-146 and described in Table 13-156.

Figure 13-146. P1_RX_DSCP_PRI_MAP7 Register

31	30	29	28	27	26	25	24
Reserved	PRI63			Reserved	PRI62		
		R/W-0h				R/W-0h	

23	22	21	20	19	18	17	16
Reserved	PRI61			Reserved	PRI60		
		R/W-0h				R/W-0h	

15	14	13	12	11	10	9	8
Reserved	PRI59			Reserved	PRI58		
		R/W-0h				R/W-0h	

7	6	5	4	3	2	1	0
Reserved	PRI57			Reserved	PRI56		
		R/W-0h				R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-156. P1_RX_DSCP_PRI_MAP7 Register Field Descriptions

Bit	Field	Type	Reset	Description
30-28	PRI63	R/W	0h	Priority 63 - A packet TOS of 0d63 is mapped to this received packet priority.
26-24	PRI62	R/W	0h	Priority 62 - A packet TOS of 0d62 is mapped to this received packet priority.
22-20	PRI61	R/W	0h	Priority 61 - A packet TOS of 0d61 is mapped to this received packet priority.
18-16	PRI60	R/W	0h	Priority 60 - A packet TOS of 0d60 is mapped to this received packet priority.
14-12	PRI59	R/W	0h	Priority 59 - A packet TOS of 0d59 is mapped to this received packet priority.
10-8	PRI58	R/W	0h	Priority 58 - A packet TOS of 0d58 is mapped to this received packet priority.
6-4	PRI57	R/W	0h	Priority 57 - A packet TOS of 0d57 is mapped to this received packet priority.
2-0	PRI56	R/W	0h	Priority 56 - A packet TOS of 0d56 is mapped to this received packet priority.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.26 P0_RX_DSCP_PRI_MAP0 Register (offset = 30h) [reset = 0h]

CPSW PORT 0 RX DSCP PRIORITY TO RX PACKET MAPPING REG 0

P0_RX_DSCP_PRI_MAP0 is shown in Figure 13-147 and described in Table 13-157.

Figure 13-147. P0_RX_DSCP_PRI_MAP0 Register

31	30	29	28	27	26	25	24
Reserved	PRI7			Reserved	PRI6		
	R/W-0h				R/W-0h		

23	22	21	20	19	18	17	16
Reserved	PRI5			Reserved	PRI4		
	R/W-0h				R/W-0h		

15	14	13	12	11	10	9	8
Reserved	PRI3			Reserved	PRI2		
	R/W-0h				R/W-0h		

7	6	5	4	3	2	1	0
Reserved	PRI1			Reserved	PRI0		
	R/W-0h				R/W-0h		

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-157. P0_RX_DSCP_PRI_MAP0 Register Field Descriptions

Bit	Field	Type	Reset	Description
30-28	PRI7	R/W	0h	Priority 7 - A packet TOS of 0d7 is mapped to this received packet priority.
26-24	PRI6	R/W	0h	Priority 6 - A packet TOS of 0d6 is mapped to this received packet priority.
22-20	PRI5	R/W	0h	Priority 5 - A packet TOS of 0d5 is mapped to this received packet priority.
18-16	PRI4	R/W	0h	Priority 4 - A packet TOS of 0d4 is mapped to this received packet priority.
14-12	PRI3	R/W	0h	Priority 3 - A packet TOS of 0d3 is mapped to this received packet priority.
10-8	PRI2	R/W	0h	Priority 2 - A packet TOS of 0d2 is mapped to this received packet priority.
6-4	PRI1	R/W	0h	Priority 1 - A packet TOS of 0d1 is mapped to this received packet priority.
2-0	PRI0	R/W	0h	Priority 0 - A packet TOS of 0d0 is mapped to this received packet priority.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.27 P2_CONTROL Register (offset = 200h) [reset = 0h]

CPSW_3GF PORT 2 CONTROL REGISTER

P2_CONTROL is shown in Figure 13-148 and described in Table 13-158.

Figure 13-148. P2_CONTROL Register

31	30	29	28	27	26	25	24
Reserved							P2_PASS_PRI_TAGGED
							R/W-0h

23	22	21	20	19	18	17	16
Reserved		P2_VLAN_LTYPE2_EN	P2_VLAN_LTYPE1_EN	P2_TS_129	Reserved		P2_DSCP_PRI_EN
		R/W-0h	R/W-0h	R/W-0h			R/W-0h

15	14	13	12	11	10	9	8
Reserved	P2_TS_320	P2_TS_319	P2_TS_132	P2_TS_131	P2_TS_130	Reserved	
	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h		

7	6	5	4	3	2	1	0
Reserved			P2_TS_ANNEX_D_EN	P2_TS_LTYPE2_EN	P2_TS_LTYPE1_EN	P2_TS_TX_EN	P2_TS_RX_EN
			R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-158. P2_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
24	P2_PASS_PRI_TAGGED	R/W	0h	Port 2 Pass Priority Tagged 0 - Priority tagged packets have the zero VID replaced with the input port P2_PORT_VLAN[11:0] 1 - Priority tagged packets are processed unchanged.
21	P2_VLAN_LTYPE2_EN	R/W	0h	Port 2 VLAN LTYPE 2 enable 0 - disabled 1 - VLAN LTYPE2 enabled on transmit and receive
20	P2_VLAN_LTYPE1_EN	R/W	0h	Port 2 VLAN LTYPE 1 enable 0 - disabled 1 - VLAN LTYPE1 enabled on transmit and receive
19	P2_TS_129	R/W	0h	Port 2 Time Sync Destination IP Address 129 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination IP address number 129 (decimal) is enabled.
16	P2_DSCP_PRI_EN	R/W	0h	Port 0 DSCP Priority Enable 0 - DSCP priority disabled 1 - DSCP priority enabled. All non-tagged IPV4 packets have their received packet priority determined by mapping the 6 TOS bits through the port DSCP priority mapping registers.
14	P2_TS_320	R/W	0h	Port 2 Time Sync Destination Port Number 320 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination port number 320 (decimal) is enabled.
13	P2_TS_319	R/W	0h	Port 2 Time Sync Destination Port Number 319 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination port number 319 (decimal) is enabled.
12	P2_TS_132	R/W	0h	Port 2 Time Sync Destination IP Address 132 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination IP address number 132 (decimal) is enabled.
11	P2_TS_131	R/W	0h	Port 2 Time Sync Destination IP Address 131 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination IP address number 131 (decimal) is enabled.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 13-158. P2_CONTROL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
10	P2_TS_130	R/W	0h	Port 2 Time Sync Destination IP Address 130 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination IP address number 130 (decimal) is enabled.
4	P2_TS_ANNEX_D_EN	R/W	0h	Port 2 Time Sync Annex D enable 0 - Annex D disabled 1 - Annex D enabled
3	P2_TS_LTYPE2_EN	R	0h	Port 2 Time Sync LTYPE 2 enable 0 - disabled 1 - enabled
2	P2_TS_LTYPE1_EN	R/W	0h	Port 2 Time Sync LTYPE 1 enable 0 - disabled 1 - enabled
1	P2_TS_TX_EN	R/W	0h	Port 2 Time Sync Transmit Enable 0 - disabled 1 - enabled
0	P2_TS_RX_EN	R/W	0h	Port 2 Time Sync Receive Enable 0 - Port 1 Receive Time Sync disabled 1 - Port 1 Receive Time Sync enabled

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.28 P0_RX_DSCP_PRI_MAP1 Register (offset = 34h) [reset = 0h]

CPSW PORT 0 RX DSCP PRIORITY TO RX PACKET MAPPING REG 1

P0_RX_DSCP_PRI_MAP1 is shown in Figure 13-149 and described in Table 13-159.

Figure 13-149. P0_RX_DSCP_PRI_MAP1 Register

31	30	29	28	27	26	25	24
Reserved	PRI15			Reserved	PRI14		
		R/W-0h				R/W-0h	

23	22	21	20	19	18	17	16
Reserved	PRI13			Reserved	PRI12		
		R/W-0h				R/W-0h	

15	14	13	12	11	10	9	8
Reserved	PRI11			Reserved	PRI10		
		R/W-0h				R/W-0h	

7	6	5	4	3	2	1	0
Reserved	PRI9			Reserved	PRI8		
		R/W-0h				R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-159. P0_RX_DSCP_PRI_MAP1 Register Field Descriptions

Bit	Field	Type	Reset	Description
30-28	PRI15	R/W	0h	Priority 15 - A packet TOS of 0d15 is mapped to this received packet priority.
26-24	PRI14	R/W	0h	Priority 14 - A packet TOS of 0d14 is mapped to this received packet priority.
22-20	PRI13	R/W	0h	Priority 13 - A packet TOS of 0d13 is mapped to this received packet priority.
18-16	PRI12	R/W	0h	Priority 12 - A packet TOS of 0d12 is mapped to this received packet priority.
14-12	PRI11	R/W	0h	Priority 11 - A packet TOS of 0d11 is mapped to this received packet priority.
10-8	PRI10	R/W	0h	Priority 10 - A packet TOS of 0d10 is mapped to this received packet priority.
6-4	PRI9	R/W	0h	Priority 9 - A packet TOS of 0d9 is mapped to this received packet priority.
2-0	PRI8	R/W	0h	Priority 8 - A packet TOS of 0d8 is mapped to this received packet priority.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.29 P2_MAX_BLKS Register (offset = 208h) [reset = 113h]

CPSW PORT 2 MAXIMUM FIFO BLOCKS REGISTER

P2_MAX_BLKS is shown in Figure 13-150 and described in Table 13-160.

Figure 13-150. P2_MAX_BLKS Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							P2_TX_MAX_BLKS
							R/W-11h

7	6	5	4	3	2	1	0
P2_TX_MAX_BLKS				P2_RX_MAX_BLKS			
R/W-11h				R/W-3h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-160. P2_MAX_BLKS Register Field Descriptions

Bit	Field	Type	Reset	Description
8-4	P2_TX_MAX_BLKS	R/W	11h	Transmit FIFO Maximum Blocks - This value is the maximum number of 1k memory blocks that may be allocated to the FIFO's logical transmit priority queues. 0x11 is the recommended value of p2_tx_max_blks unless the port is in fullduplex flow control mode. In flow control mode, the p2_rx_max_blks will need to increase in order to accept the required run out in fullduplex mode. This value will need to decrease by the amount of increase in p2_rx_max_blks. 0xe is the minimum value tx max blks.
3-0	P2_RX_MAX_BLKS	R/W	3h	Receive FIFO Maximum Blocks - This value is the maximum number of 1k memory blocks that may be allocated to the FIFO's logical receive queue. This value must be greater than or equal to 0x3. It should be increased In fullduplex flow control mode to 0x5 or 0x6 depending on the required runout space. The p2_tx_max_blks value must be decreased by the amount of increase in p2_rx_max_blks. 0x3 is the minimum value rx max blks and 0x6 is the maximum value.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.30 P2_BLK_CNT Register (offset = 20Ch) [reset = 41h]

CPSW PORT 2 FIFO BLOCK USAGE COUNT (READ ONLY)

P2_BLK_CNT is shown in Figure 13-151 and described in Table 13-161.

Figure 13-151. P2_BLK_CNT Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							P2_TX_BLK_CNT
							R-4h

7	6	5	4	3	2	1	0
P2_TX_BLK_CNT				P2_RX_BLK_CNT			
R-4h				R-1h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-161. P2_BLK_CNT Register Field Descriptions

Bit	Field	Type	Reset	Description
8-4	P2_TX_BLK_CNT	R	4h	Port 2 Transmit Block Count Usage - This value is the number of blocks allocated to the FIFO logical transmit queues.
3-0	P2_RX_BLK_CNT	R	1h	Port 2 Receive Block Count Usage - This value is the number of blocks allocated to the FIFO logical receive queues.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.31 P2_TX_IN_CTL Register (offset = 210h) [reset = 80040C0h]

CPSW PORT 2 TRANSMIT FIFO CONTROL

P2_TX_IN_CTL is shown in Figure 13-152 and described in Table 13-162.

Figure 13-152. P2_TX_IN_CTL Register

31	30	29	28	27	26	25	24
Reserved				HOST_BLKS_REM			
				R/W-8h			

23	22	21	20	19	18	17	16
TX_RATE_EN				Reserved		TX_IN_SEL	
R/W-0h						R/W-0h	

15	14	13	12	11	10	9	8
TX_BLKS_REM				Reserved		TX_PRI_WDS	
R/W-4h						R/W-C0h	

7	6	5	4	3	2	1	0
TX_PRI_WDS							
R/W-C0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-162. P2_TX_IN_CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
27-24	HOST_BLKS_REM	R/W	8h	Transmit FIFO Blocks that must be free before a non rate-limited CPDMA channel can begin sending a packet to the FIFO.
23-20	TX_RATE_EN	R/W	0h	Transmit FIFO Input Rate Enable
17-16	TX_IN_SEL	R/W	0h	Transmit FIFO Input Queue Type Select 00 - Normal priority mode 01 - reserved 10 - Rate Limit mode 11 - reserved
15-12	TX_BLKS_REM	R/W	4h	Transmit FIFO Input Blocks to subtract in dual mac mode and blocks to subtract on non rate-limited traffic in rate-limit mode.
9-0	TX_PRI_WDS	R/W	C0h	Transmit FIFO Words in queue

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.32 P2_PORT_VLAN Register (offset = 214h) [reset = 0h]

CPSW PORT 2 VLAN REGISTER

P2_PORT_VLAN is shown in Figure 13-153 and described in Table 13-163.

Figure 13-153. P2_PORT_VLAN Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
PORT_PRI			PORT_CFI	PORT_VID			
R/W-0h			R/W-0h	R/W-0h			

7	6	5	4	3	2	1	0
PORT_VID							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-163. P2_PORT_VLAN Register Field Descriptions

Bit	Field	Type	Reset	Description
15-13	PORT_PRI	R/W	0h	Port VLAN Priority (7 is highest priority)
12	PORT_CFI	R/W	0h	Port CFI bit
11-0	PORT_VID	R/W	0h	Port VLAN ID

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.33 P2_TX_PRI_MAP Register (offset = 218h) [reset = 33221001h]

CPSW PORT 2 TX HEADER PRIORITY TO SWITCH PRI MAPPING REGISTER

P2_TX_PRI_MAP is shown in Figure 13-154 and described in Table 13-164.

Figure 13-154. P2_TX_PRI_MAP Register

31	30	29	28	27	26	25	24
Reserved		PRI7		Reserved		PRI6	
		R/W-3h				R/W-3h	

23	22	21	20	19	18	17	16
Reserved		PRI5		Reserved		PRI4	
		R/W-2h				R/W-2h	

15	14	13	12	11	10	9	8
Reserved		PRI3		Reserved		PRI2	
		R/W-1h				R/W-0h	

7	6	5	4	3	2	1	0
Reserved		PRI1		Reserved		PRI0	
		R/W-0h				R/W-1h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-164. P2_TX_PRI_MAP Register Field Descriptions

Bit	Field	Type	Reset	Description
29-28	PRI7	R/W	3h	Priority 7 - A packet header priority of 0x7 is given this switch queue pri.
25-24	PRI6	R/W	3h	Priority 6 - A packet header priority of 0x6 is given this switch queue pri.
21-20	PRI5	R/W	2h	Priority 5 - A packet header priority of 0x5 is given this switch queue pri.
17-16	PRI4	R/W	2h	Priority 4 - A packet header priority of 0x4 is given this switch queue pri.
13-12	PRI3	R/W	1h	Priority 3 - A packet header priority of 0x3 is given this switch queue pri.
9-8	PRI2	R/W	0h	Priority 2 - A packet header priority of 0x2 is given this switch queue pri.
5-4	PRI1	R/W	0h	Priority 1 - A packet header priority of 0x1 is given this switch queue pri.
1-0	PRI0	R/W	1h	Priority 0 - A packet header priority of 0x0 is given this switch queue pri.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.34 P2_TS_SEQ_MTYPE Register (offset = 21Ch) [reset = 1E0000h]

CPSW_3GF PORT 2 TIME SYNC SEQUENCE ID OFFSET AND MSG TYPE.

P2_TS_SEQ_MTYPE is shown in Figure 13-155 and described in Table 13-165.

Figure 13-155. P2_TS_SEQ_MTYPE Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved		P2_TS_SEQ_ID_OFFSET					
		R/W-1Eh					

15	14	13	12	11	10	9	8
P2_TS_MSG_TYPE_EN							
R/W-0h							

7	6	5	4	3	2	1	0
P2_TS_MSG_TYPE_EN							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-165. P2_TS_SEQ_MTYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
21-16	P2_TS_SEQ_ID_OFFSET	R/W	1Eh	Port 2 Time Sync Sequence ID Offset This is the number of octets that the sequence ID is offset in the tx and rx time sync message header. The minimum value is 6.
15-0	P2_TS_MSG_TYPE_EN	R/W	0h	Port 2 Time Sync Message Type Enable - Each bit in this field enables the corresponding message type in receive and transmit time sync messages (Bit 0 enables message type 0 etc. .

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.35 P2_SA_LO Register (offset = 220h) [reset = 0h]

CPSW CPGMAC_SL2 SOURCE ADDRESS LOW REGISTER

P2_SA_LO is shown in Figure 13-156 and described in Table 13-166.

Figure 13-156. P2_SA_LO Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
MACSRCADDR_7_0							
R/W-0h							

7	6	5	4	3	2	1	0
MACSRCADDR_15_8							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-166. P2_SA_LO Register Field Descriptions

Bit	Field	Type	Reset	Description
15-8	MACSRCADDR_7_0	R/W	0h	Source Address Lower 8 bits (byte 0)
7-0	MACSRCADDR_15_8	R/W	0h	Source Address bits 15:8 (byte 1)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.36 P2_SA_HI Register (offset = 224h) [reset = 0h]

CPSW CPGMAC_SL2 SOURCE ADDRESS HIGH REGISTER

P2_SA_HI is shown in Figure 13-157 and described in Table 13-167.

Figure 13-157. P2_SA_HI Register

31	30	29	28	27	26	25	24
MACSRCADDR_23_16							
R/W-0h							

23	22	21	20	19	18	17	16
MACSRCADDR_31_23							
R/W-0h							

15	14	13	12	11	10	9	8
MACSRCADDR_39_32							
R/W-0h							

7	6	5	4	3	2	1	0
MACSRCADDR_47_40							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-167. P2_SA_HI Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	MACSRCADDR_23_16	R/W	0h	Source Address bits 23:16 (byte 2)
23-16	MACSRCADDR_31_23	R/W	0h	Source Address bits 31:23 (byte 3)
15-8	MACSRCADDR_39_32	R/W	0h	Source Address bits 39:32 (byte 4)
7-0	MACSRCADDR_47_40	R/W	0h	Source Address bits 47:40 (byte 5)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.37 P2_SEND_PERCENT Register (offset = 228h) [reset = 0h]

CPSW PORT 2 TRANSMIT QUEUE SEND PERCENTAGES

P2_SEND_PERCENT is shown in Figure 13-158 and described in Table 13-168.

Figure 13-158. P2_SEND_PERCENT Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved	PRI3_SEND_PERCENT						
	R/W-0h						

15	14	13	12	11	10	9	8
Reserved	PRI2_SEND_PERCENT						
	R/W-0h						

7	6	5	4	3	2	1	0
Reserved	PRI1_SEND_PERCENT						
	R/W-0h						

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-168. P2_SEND_PERCENT Register Field Descriptions

Bit	Field	Type	Reset	Description
22-16	PRI3_SEND_PERCENT	R/W	0h	Priority 3 Transmit Percentage - This percentage value is sent from FIFO priority 3 (maximum) when the p1_pri3_shape_en is set (queue shaping enabled). This is the percentage of the wire that packets from priority 3 receive (which includes interpacket gap and preamble bytes). If shaping is enabled on this queue then this value must be between zero and 0d100 (not inclusive).
14-8	PRI2_SEND_PERCENT	R/W	0h	Priority 2 Transmit Percentage - This percentage value is sent from FIFO priority 2 (maximum) when the p1_pri2_shape_en is set (queue shaping enabled). This is the percentage of the wire that packets from priority 2 receive (which includes interpacket gap and preamble bytes). If shaping is enabled on this queue then this value must be between zero and 0d100 (not inclusive).
6-0	PRI1_SEND_PERCENT	R/W	0h	Priority 1 Transmit Percentage - This percentage value is sent from FIFO priority 1 (maximum) when the p1_pri1_shape_en is set (queue shaping enabled). This is the percentage of the wire that packets from priority 1 receive (which includes interpacket gap and preamble bytes). If shaping is enabled on this queue then this value must be between zero and 0d100 (not inclusive).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.38 P0_RX_DSCP_PRI_MAP2 Register (offset = 38h) [reset = 0h]

CPSW PORT 0 RX DSCP PRIORITY TO RX PACKET MAPPING REG 2

P0_RX_DSCP_PRI_MAP2 is shown in Figure 13-159 and described in Table 13-169.

Figure 13-159. P0_RX_DSCP_PRI_MAP2 Register

31	30	29	28	27	26	25	24
Reserved	PRI23			Reserved	PRI22		
		R/W-0h				R/W-0h	

23	22	21	20	19	18	17	16
Reserved	PRI21			Reserved	PRI20		
		R/W-0h				R/W-0h	

15	14	13	12	11	10	9	8
Reserved	PRI19			Reserved	PRI18		
		R/W-0h				R/W-0h	

7	6	5	4	3	2	1	0
Reserved	PRI17			Reserved	PRI16		
		R/W-0h				R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-169. P0_RX_DSCP_PRI_MAP2 Register Field Descriptions

Bit	Field	Type	Reset	Description
30-28	PRI23	R/W	0h	Priority 23 - A packet TOS of 0d23 is mapped to this received packet priority.
26-24	PRI22	R/W	0h	Priority 22 - A packet TOS of 0d22 is mapped to this received packet priority.
22-20	PRI21	R/W	0h	Priority 21 - A packet TOS of 0d21 is mapped to this received packet priority.
18-16	PRI20	R/W	0h	Priority 20 - A packet TOS of 0d20 is mapped to this received packet priority.
14-12	PRI19	R/W	0h	Priority 19 - A packet TOS of 0d19 is mapped to this received packet priority.
10-8	PRI18	R/W	0h	Priority 18 - A packet TOS of 0d18 is mapped to this received packet priority.
6-4	PRI17	R/W	0h	Priority 17 - A packet TOS of 0d17 is mapped to this received packet priority.
2-0	PRI16	R/W	0h	Priority 16 - A packet TOS of 0d16 is mapped to this received packet priority.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.39 P2_RX_DSCP_PRI_MAP0 Register (offset = 230h) [reset = 0h]

CPSW PORT 2 RX DSCP PRIORITY TO RX PACKET MAPPING REG 0

P2_RX_DSCP_PRI_MAP0 is shown in Figure 13-160 and described in Table 13-170.

Figure 13-160. P2_RX_DSCP_PRI_MAP0 Register

31	30	29	28	27	26	25	24
Reserved	PRI7			Reserved	PRI6		
	R/W-0h				R/W-0h		

23	22	21	20	19	18	17	16
Reserved	PRI5			Reserved	PRI4		
	R/W-0h				R/W-0h		

15	14	13	12	11	10	9	8
Reserved	PRI3			Reserved	PRI2		
	R/W-0h				R/W-0h		

7	6	5	4	3	2	1	0
Reserved	PRI1			Reserved	PRI0		
	R/W-0h				R/W-0h		

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-170. P2_RX_DSCP_PRI_MAP0 Register Field Descriptions

Bit	Field	Type	Reset	Description
30-28	PRI7	R/W	0h	Priority 7 - A packet TOS of 0d7 is mapped to this received packet priority.
26-24	PRI6	R/W	0h	Priority 6 - A packet TOS of 0d6 is mapped to this received packet priority.
22-20	PRI5	R/W	0h	Priority 5 - A packet TOS of 0d5 is mapped to this received packet priority.
18-16	PRI4	R/W	0h	Priority 4 - A packet TOS of 0d4 is mapped to this received packet priority.
14-12	PRI3	R/W	0h	Priority 3 - A packet TOS of 0d3 is mapped to this received packet priority.
10-8	PRI2	R/W	0h	Priority 2 - A packet TOS of 0d2 is mapped to this received packet priority.
6-4	PRI1	R/W	0h	Priority 1 - A packet TOS of 0d1 is mapped to this received packet priority.
2-0	PRI0	R/W	0h	Priority 0 - A packet TOS of 0d0 is mapped to this received packet priority.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.40 P2_RX_DSCP_PRI_MAP1 Register (offset = 234h) [reset = 0h]

CPSW PORT 2 RX DSCP PRIORITY TO RX PACKET MAPPING REG 1

P2_RX_DSCP_PRI_MAP1 is shown in Figure 13-161 and described in Table 13-171.

Figure 13-161. P2_RX_DSCP_PRI_MAP1 Register

31	30	29	28	27	26	25	24
Reserved	PRI15			Reserved	PRI14		
		R/W-0h				R/W-0h	

23	22	21	20	19	18	17	16
Reserved	PRI13			Reserved	PRI12		
		R/W-0h				R/W-0h	

15	14	13	12	11	10	9	8
Reserved	PRI11			Reserved	PRI10		
		R/W-0h				R/W-0h	

7	6	5	4	3	2	1	0
Reserved	PRI9			Reserved	PRI8		
		R/W-0h				R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-171. P2_RX_DSCP_PRI_MAP1 Register Field Descriptions

Bit	Field	Type	Reset	Description
30-28	PRI15	R/W	0h	Priority 15 - A packet TOS of 0d15 is mapped to this received packet priority.
26-24	PRI14	R/W	0h	Priority 14 - A packet TOS of 0d14 is mapped to this received packet priority.
22-20	PRI13	R/W	0h	Priority 13 - A packet TOS of 0d13 is mapped to this received packet priority.
18-16	PRI12	R/W	0h	Priority 12 - A packet TOS of 0d12 is mapped to this received packet priority.
14-12	PRI11	R/W	0h	Priority 11 - A packet TOS of 0d11 is mapped to this received packet priority.
10-8	PRI10	R/W	0h	Priority 10 - A packet TOS of 0d10 is mapped to this received packet priority.
6-4	PRI9	R/W	0h	Priority 9 - A packet TOS of 0d9 is mapped to this received packet priority.
2-0	PRI8	R/W	0h	Priority 8 - A packet TOS of 0d8 is mapped to this received packet priority.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.41 P2_RX_DSCP_PRI_MAP2 Register (offset = 238h) [reset = 0h]

CPSW PORT 2 RX DSCP PRIORITY TO RX PACKET MAPPING REG 2

P2_RX_DSCP_PRI_MAP2 is shown in Figure 13-162 and described in Table 13-172.

Figure 13-162. P2_RX_DSCP_PRI_MAP2 Register

31	30	29	28	27	26	25	24
Reserved	PRI23			Reserved	PRI22		
		R/W-0h				R/W-0h	

23	22	21	20	19	18	17	16
Reserved	PRI21			Reserved	PRI20		
		R/W-0h				R/W-0h	

15	14	13	12	11	10	9	8
Reserved	PRI19			Reserved	PRI18		
		R/W-0h				R/W-0h	

7	6	5	4	3	2	1	0
Reserved	PRI17			Reserved	PRI16		
		R/W-0h				R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-172. P2_RX_DSCP_PRI_MAP2 Register Field Descriptions

Bit	Field	Type	Reset	Description
30-28	PRI23	R/W	0h	Priority 23 - A packet TOS of 0d23 is mapped to this received packet priority.
26-24	PRI22	R/W	0h	Priority 22 - A packet TOS of 0d22 is mapped to this received packet priority.
22-20	PRI21	R/W	0h	Priority 21 - A packet TOS of 0d21 is mapped to this received packet priority.
18-16	PRI20	R/W	0h	Priority 20 - A packet TOS of 0d20 is mapped to this received packet priority.
14-12	PRI19	R/W	0h	Priority 19 - A packet TOS of 0d19 is mapped to this received packet priority.
10-8	PRI18	R/W	0h	Priority 18 - A packet TOS of 0d18 is mapped to this received packet priority.
6-4	PRI17	R/W	0h	Priority 17 - A packet TOS of 0d17 is mapped to this received packet priority.
2-0	PRI16	R/W	0h	Priority 16 - A packet TOS of 0d16 is mapped to this received packet priority.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.42 P2_RX_DSCP_PRI_MAP3 Register (offset = 23Ch) [reset = 0h]

CPSW PORT 2 RX DSCP PRIORITY TO RX PACKET MAPPING REG 3

P2_RX_DSCP_PRI_MAP3 is shown in Figure 13-163 and described in Table 13-173.

Figure 13-163. P2_RX_DSCP_PRI_MAP3 Register

31	30	29	28	27	26	25	24
Reserved	PRI31			Reserved	PRI30		
		R/W-0h				R/W-0h	

23	22	21	20	19	18	17	16
Reserved	PRI29			Reserved	PRI28		
		R/W-0h				R/W-0h	

15	14	13	12	11	10	9	8
Reserved	PRI27			Reserved	PRI26		
		R/W-0h				R/W-0h	

7	6	5	4	3	2	1	0
Reserved	PRI25			Reserved	PRI24		
		R/W-0h				R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-173. P2_RX_DSCP_PRI_MAP3 Register Field Descriptions

Bit	Field	Type	Reset	Description
30-28	PRI31	R/W	0h	Priority 31 - A packet TOS of 0d31 is mapped to this received packet priority.
26-24	PRI30	R/W	0h	Priority 30 - A packet TOS of 0d30 is mapped to this received packet priority.
22-20	PRI29	R/W	0h	Priority 29 - A packet TOS of 0d39 is mapped to this received packet priority.
18-16	PRI28	R/W	0h	Priority 28 - A packet TOS of 0d28 is mapped to this received packet priority.
14-12	PRI27	R/W	0h	Priority 27 - A packet TOS of 0d27 is mapped to this received packet priority.
10-8	PRI26	R/W	0h	Priority 26 - A packet TOS of 0d26 is mapped to this received packet priority.
6-4	PRI25	R/W	0h	Priority 25 - A packet TOS of 0d25 is mapped to this received packet priority.
2-0	PRI24	R/W	0h	Priority 24 - A packet TOS of 0d24 is mapped to this received packet priority.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.43 P2_RX_DSCP_PRI_MAP4 Register (offset = 240h) [reset = 0h]

CPSW PORT 2 RX DSCP PRIORITY TO RX PACKET MAPPING REG 4

P2_RX_DSCP_PRI_MAP4 is shown in Figure 13-164 and described in Table 13-174.

Figure 13-164. P2_RX_DSCP_PRI_MAP4 Register

31	30	29	28	27	26	25	24
Reserved	PRI39			Reserved	PRI38		
		R/W-0h				R/W-0h	

23	22	21	20	19	18	17	16
Reserved	PRI37			Reserved	PRI36		
		R/W-0h				R/W-0h	

15	14	13	12	11	10	9	8
Reserved	PRI35			Reserved	PRI34		
		R/W-0h				R/W-0h	

7	6	5	4	3	2	1	0
Reserved	PRI33			Reserved	PRI32		
		R/W-0h				R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-174. P2_RX_DSCP_PRI_MAP4 Register Field Descriptions

Bit	Field	Type	Reset	Description
30-28	PRI39	R/W	0h	Priority 39 - A packet TOS of 0d39 is mapped to this received packet priority.
26-24	PRI38	R/W	0h	Priority 38 - A packet TOS of 0d38 is mapped to this received packet priority.
22-20	PRI37	R/W	0h	Priority 37 - A packet TOS of 0d37 is mapped to this received packet priority.
18-16	PRI36	R/W	0h	Priority 36 - A packet TOS of 0d36 is mapped to this received packet priority.
14-12	PRI35	R/W	0h	Priority 35 - A packet TOS of 0d35 is mapped to this received packet priority.
10-8	PRI34	R/W	0h	Priority 34 - A packet TOS of 0d34 is mapped to this received packet priority.
6-4	PRI33	R/W	0h	Priority 33 - A packet TOS of 0d33 is mapped to this received packet priority.
2-0	PRI32	R/W	0h	Priority 32 - A packet TOS of 0d32 is mapped to this received packet priority.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.44 P2_RX_DSCP_PRI_MAP5 Register (offset = 244h) [reset = 0h]

CPSW PORT 2 RX DSCP PRIORITY TO RX PACKET MAPPING REG 5

P2_RX_DSCP_PRI_MAP5 is shown in Figure 13-165 and described in Table 13-175.

Figure 13-165. P2_RX_DSCP_PRI_MAP5 Register

31	30	29	28	27	26	25	24
Reserved	PRI47			Reserved	PRI46		
		R/W-0h				R/W-0h	

23	22	21	20	19	18	17	16
Reserved	PRI45			Reserved	PRI44		
		R/W-0h				R/W-0h	

15	14	13	12	11	10	9	8
Reserved	PRI43			Reserved	PRI42		
		R/W-0h				R/W-0h	

7	6	5	4	3	2	1	0
Reserved	PRI41			Reserved	PRI40		
		R/W-0h				R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-175. P2_RX_DSCP_PRI_MAP5 Register Field Descriptions

Bit	Field	Type	Reset	Description
30-28	PRI47	R/W	0h	Priority 47 - A packet TOS of 0d47 is mapped to this received packet priority.
26-24	PRI46	R/W	0h	Priority 46 - A packet TOS of 0d46 is mapped to this received packet priority.
22-20	PRI45	R/W	0h	Priority 45 - A packet TOS of 0d45 is mapped to this received packet priority.
18-16	PRI44	R/W	0h	Priority 44 - A packet TOS of 0d44 is mapped to this received packet priority.
14-12	PRI43	R/W	0h	Priority 43 - A packet TOS of 0d43 is mapped to this received packet priority.
10-8	PRI42	R/W	0h	Priority 42 - A packet TOS of 0d42 is mapped to this received packet priority.
6-4	PRI41	R/W	0h	Priority 41 - A packet TOS of 0d41 is mapped to this received packet priority.
2-0	PRI40	R/W	0h	Priority 40 - A packet TOS of 0d40 is mapped to this received packet priority.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.45 P2_RX_DSCP_PRI_MAP6 Register (offset = 248h) [reset = 0h]

CPSW PORT 2 RX DSCP PRIORITY TO RX PACKET MAPPING REG 6

P2_RX_DSCP_PRI_MAP6 is shown in Figure 13-166 and described in Table 13-176.

Figure 13-166. P2_RX_DSCP_PRI_MAP6 Register

31	30	29	28	27	26	25	24
Reserved	PRI55			Reserved	PRI54		
		R/W-0h				R/W-0h	

23	22	21	20	19	18	17	16
Reserved	PRI53			Reserved	PRI52		
		R/W-0h				R/W-0h	

15	14	13	12	11	10	9	8
Reserved	PRI51			Reserved	PRI50		
		R/W-0h				R/W-0h	

7	6	5	4	3	2	1	0
Reserved	PRI49			Reserved	PRI48		
		R/W-0h				R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-176. P2_RX_DSCP_PRI_MAP6 Register Field Descriptions

Bit	Field	Type	Reset	Description
30-28	PRI55	R/W	0h	Priority 55 - A packet TOS of 0d55 is mapped to this received packet priority.
26-24	PRI54	R/W	0h	Priority 54 - A packet TOS of 0d54 is mapped to this received packet priority.
22-20	PRI53	R/W	0h	Priority 53 - A packet TOS of 0d53 is mapped to this received packet priority.
18-16	PRI52	R/W	0h	Priority 52 - A packet TOS of 0d52 is mapped to this received packet priority.
14-12	PRI51	R/W	0h	Priority 51 - A packet TOS of 0d51 is mapped to this received packet priority.
10-8	PRI50	R/W	0h	Priority 50 - A packet TOS of 0d50 is mapped to this received packet priority.
6-4	PRI49	R/W	0h	Priority 49 - A packet TOS of 0d49 is mapped to this received packet priority.
2-0	PRI48	R/W	0h	Priority 48 - A packet TOS of 0d48 is mapped to this received packet priority.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.46 P2_RX_DSCP_PRI_MAP7 Register (offset = 24Ch) [reset = 0h]

CPSW PORT 2 RX DSCP PRIORITY TO RX PACKET MAPPING REG 7

P2_RX_DSCP_PRI_MAP7 is shown in Figure 13-167 and described in Table 13-177.

Figure 13-167. P2_RX_DSCP_PRI_MAP7 Register

31	30	29	28	27	26	25	24
Reserved	PRI63			Reserved	PRI62		
		R/W-0h				R/W-0h	

23	22	21	20	19	18	17	16
Reserved	PRI61			Reserved	PRI60		
		R/W-0h				R/W-0h	

15	14	13	12	11	10	9	8
Reserved	PRI59			Reserved	PRI58		
		R/W-0h				R/W-0h	

7	6	5	4	3	2	1	0
Reserved	PRI57			Reserved	PRI56		
		R/W-0h				R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-177. P2_RX_DSCP_PRI_MAP7 Register Field Descriptions

Bit	Field	Type	Reset	Description
30-28	PRI63	R/W	0h	Priority 63 - A packet TOS of 0d63 is mapped to this received packet priority.
26-24	PRI62	R/W	0h	Priority 62 - A packet TOS of 0d62 is mapped to this received packet priority.
22-20	PRI61	R/W	0h	Priority 61 - A packet TOS of 0d61 is mapped to this received packet priority.
18-16	PRI60	R/W	0h	Priority 60 - A packet TOS of 0d60 is mapped to this received packet priority.
14-12	PRI59	R/W	0h	Priority 59 - A packet TOS of 0d59 is mapped to this received packet priority.
10-8	PRI58	R/W	0h	Priority 58 - A packet TOS of 0d58 is mapped to this received packet priority.
6-4	PRI57	R/W	0h	Priority 57 - A packet TOS of 0d57 is mapped to this received packet priority.
2-0	PRI56	R/W	0h	Priority 56 - A packet TOS of 0d56 is mapped to this received packet priority.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.47 P0_RX_DSCP_PRI_MAP3 Register (offset = 3Ch) [reset = 0h]

CPSW PORT 0 RX DSCP PRIORITY TO RX PACKET MAPPING REG 3

P0_RX_DSCP_PRI_MAP3 is shown in Figure 13-168 and described in Table 13-178.

Figure 13-168. P0_RX_DSCP_PRI_MAP3 Register

31	30	29	28	27	26	25	24
Reserved	PRI31			Reserved	PRI30		
		R/W-0h				R/W-0h	

23	22	21	20	19	18	17	16
Reserved	PRI29			Reserved	PRI28		
		R/W-0h				R/W-0h	

15	14	13	12	11	10	9	8
Reserved	PRI27			Reserved	PRI26		
		R/W-0h				R/W-0h	

7	6	5	4	3	2	1	0
Reserved	PRI25			Reserved	PRI24		
		R/W-0h				R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-178. P0_RX_DSCP_PRI_MAP3 Register Field Descriptions

Bit	Field	Type	Reset	Description
30-28	PRI31	R/W	0h	Priority 31 - A packet TOS of 0d31 is mapped to this received packet priority.
26-24	PRI30	R/W	0h	Priority 30 - A packet TOS of 0d30 is mapped to this received packet priority.
22-20	PRI29	R/W	0h	Priority 29 - A packet TOS of 0d39 is mapped to this received packet priority.
18-16	PRI28	R/W	0h	Priority 28 - A packet TOS of 0d28 is mapped to this received packet priority.
14-12	PRI27	R/W	0h	Priority 27 - A packet TOS of 0d27 is mapped to this received packet priority.
10-8	PRI26	R/W	0h	Priority 26 - A packet TOS of 0d26 is mapped to this received packet priority.
6-4	PRI25	R/W	0h	Priority 25 - A packet TOS of 0d25 is mapped to this received packet priority.
2-0	PRI24	R/W	0h	Priority 24 - A packet TOS of 0d24 is mapped to this received packet priority.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.48 P0_RX_DSCP_PRI_MAP4 Register (offset = 40h) [reset = 0h]

CPSW PORT 0 RX DSCP PRIORITY TO RX PACKET MAPPING REG 4

P0_RX_DSCP_PRI_MAP4 is shown in Figure 13-169 and described in Table 13-179.

Figure 13-169. P0_RX_DSCP_PRI_MAP4 Register

31	30	29	28	27	26	25	24
Reserved	PRI39			Reserved	PRI38		
		R/W-0h				R/W-0h	

23	22	21	20	19	18	17	16
Reserved	PRI37			Reserved	PRI36		
		R/W-0h				R/W-0h	

15	14	13	12	11	10	9	8
Reserved	PRI35			Reserved	PRI34		
		R/W-0h				R/W-0h	

7	6	5	4	3	2	1	0
Reserved	PRI33			Reserved	PRI32		
		R/W-0h				R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-179. P0_RX_DSCP_PRI_MAP4 Register Field Descriptions

Bit	Field	Type	Reset	Description
30-28	PRI39	R/W	0h	Priority 39 - A packet TOS of 0d39 is mapped to this received packet priority.
26-24	PRI38	R/W	0h	Priority 38 - A packet TOS of 0d38 is mapped to this received packet priority.
22-20	PRI37	R/W	0h	Priority 37 - A packet TOS of 0d37 is mapped to this received packet priority.
18-16	PRI36	R/W	0h	Priority 36 - A packet TOS of 0d36 is mapped to this received packet priority.
14-12	PRI35	R/W	0h	Priority 35 - A packet TOS of 0d35 is mapped to this received packet priority.
10-8	PRI34	R/W	0h	Priority 34 - A packet TOS of 0d34 is mapped to this received packet priority.
6-4	PRI33	R/W	0h	Priority 33 - A packet TOS of 0d33 is mapped to this received packet priority.
2-0	PRI32	R/W	0h	Priority 32 - A packet TOS of 0d32 is mapped to this received packet priority.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.49 P0_RX_DSCP_PRI_MAP5 Register (offset = 44h) [reset = 0h]

CPSW PORT 0 RX DSCP PRIORITY TO RX PACKET MAPPING REG 5

P0_RX_DSCP_PRI_MAP5 is shown in Figure 13-170 and described in Table 13-180.

Figure 13-170. P0_RX_DSCP_PRI_MAP5 Register

31	30	29	28	27	26	25	24
Reserved	PRI47			Reserved	PRI46		
		R/W-0h				R/W-0h	

23	22	21	20	19	18	17	16
Reserved	PRI45			Reserved	PRI44		
		R/W-0h				R/W-0h	

15	14	13	12	11	10	9	8
Reserved	PRI43			Reserved	PRI42		
		R/W-0h				R/W-0h	

7	6	5	4	3	2	1	0
Reserved	PRI41			Reserved	PRI40		
		R/W-0h				R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-180. P0_RX_DSCP_PRI_MAP5 Register Field Descriptions

Bit	Field	Type	Reset	Description
30-28	PRI47	R/W	0h	Priority 47 - A packet TOS of 0d47 is mapped to this received packet priority.
26-24	PRI46	R/W	0h	Priority 46 - A packet TOS of 0d46 is mapped to this received packet priority.
22-20	PRI45	R/W	0h	Priority 45 - A packet TOS of 0d45 is mapped to this received packet priority.
18-16	PRI44	R/W	0h	Priority 44 - A packet TOS of 0d44 is mapped to this received packet priority.
14-12	PRI43	R/W	0h	Priority 43 - A packet TOS of 0d43 is mapped to this received packet priority.
10-8	PRI42	R/W	0h	Priority 42 - A packet TOS of 0d42 is mapped to this received packet priority.
6-4	PRI41	R/W	0h	Priority 41 - A packet TOS of 0d41 is mapped to this received packet priority.
2-0	PRI40	R/W	0h	Priority 40 - A packet TOS of 0d40 is mapped to this received packet priority.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.50 P0_RX_DSCP_PRI_MAP6 Register (offset = 48h) [reset = 0h]

CPSW PORT 0 RX DSCP PRIORITY TO RX PACKET MAPPING REG 6

P0_RX_DSCP_PRI_MAP6 is shown in Figure 13-171 and described in Table 13-181.

Figure 13-171. P0_RX_DSCP_PRI_MAP6 Register

31	30	29	28	27	26	25	24
Reserved	PRI55			Reserved	PRI54		
		R/W-0h				R/W-0h	

23	22	21	20	19	18	17	16
Reserved	PRI53			Reserved	PRI52		
		R/W-0h				R/W-0h	

15	14	13	12	11	10	9	8
Reserved	PRI51			Reserved	PRI50		
		R/W-0h				R/W-0h	

7	6	5	4	3	2	1	0
Reserved	PRI49			Reserved	PRI48		
		R/W-0h				R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-181. P0_RX_DSCP_PRI_MAP6 Register Field Descriptions

Bit	Field	Type	Reset	Description
30-28	PRI55	R/W	0h	Priority 55 - A packet TOS of 0d55 is mapped to this received packet priority.
26-24	PRI54	R/W	0h	Priority 54 - A packet TOS of 0d54 is mapped to this received packet priority.
22-20	PRI53	R/W	0h	Priority 53 - A packet TOS of 0d53 is mapped to this received packet priority.
18-16	PRI52	R/W	0h	Priority 52 - A packet TOS of 0d52 is mapped to this received packet priority.
14-12	PRI51	R/W	0h	Priority 51 - A packet TOS of 0d51 is mapped to this received packet priority.
10-8	PRI50	R/W	0h	Priority 50 - A packet TOS of 0d50 is mapped to this received packet priority.
6-4	PRI49	R/W	0h	Priority 49 - A packet TOS of 0d49 is mapped to this received packet priority.
2-0	PRI48	R/W	0h	Priority 48 - A packet TOS of 0d48 is mapped to this received packet priority.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.51 P0_RX_DSCP_PRI_MAP7 Register (offset = 4Ch) [reset = 0h]

CPSW PORT 0 RX DSCP PRIORITY TO RX PACKET MAPPING REG 7

P0_RX_DSCP_PRI_MAP7 is shown in Figure 13-172 and described in Table 13-182.

Figure 13-172. P0_RX_DSCP_PRI_MAP7 Register

31	30	29	28	27	26	25	24
Reserved	PRI63			Reserved	PRI62		
		R/W-0h				R/W-0h	

23	22	21	20	19	18	17	16
Reserved	PRI61			Reserved	PRI60		
		R/W-0h				R/W-0h	

15	14	13	12	11	10	9	8
Reserved	PRI59			Reserved	PRI58		
		R/W-0h				R/W-0h	

7	6	5	4	3	2	1	0
Reserved	PRI57			Reserved	PRI56		
		R/W-0h				R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-182. P0_RX_DSCP_PRI_MAP7 Register Field Descriptions

Bit	Field	Type	Reset	Description
30-28	PRI63	R/W	0h	Priority 63 - A packet TOS of 0d63 is mapped to this received packet priority.
26-24	PRI62	R/W	0h	Priority 62 - A packet TOS of 0d62 is mapped to this received packet priority.
22-20	PRI61	R/W	0h	Priority 61 - A packet TOS of 0d61 is mapped to this received packet priority.
18-16	PRI60	R/W	0h	Priority 60 - A packet TOS of 0d60 is mapped to this received packet priority.
14-12	PRI59	R/W	0h	Priority 59 - A packet TOS of 0d59 is mapped to this received packet priority.
10-8	PRI58	R/W	0h	Priority 58 - A packet TOS of 0d58 is mapped to this received packet priority.
6-4	PRI57	R/W	0h	Priority 57 - A packet TOS of 0d57 is mapped to this received packet priority.
2-0	PRI56	R/W	0h	Priority 56 - A packet TOS of 0d56 is mapped to this received packet priority.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.8.52 P0_MAX_BLKS Register (offset = 8h) [reset = 104h]

CPSW PORT 0 MAXIMUM FIFO BLOCKS REGISTER

P0_MAX_BLKS is shown in Figure 13-173 and described in Table 13-183.

Figure 13-173. P0_MAX_BLKS Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							P0_TX_MAX_BLKS
							R/W-10h

7	6	5	4	3	2	1	0
P0_TX_MAX_BLKS				P0_RX_MAX_BLKS			
R/W-10h				R/W-4h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-183. P0_MAX_BLKS Register Field Descriptions

Bit	Field	Type	Reset	Description
8-4	P0_TX_MAX_BLKS	R/W	10h	Transmit FIFO Maximum Blocks - This value is the maximum number of 1k memory blocks that may be allocated to the FIFO's logical transmit priority queues. 0x10 is the recommended value of p0_tx_max_blks. Port 0 should remain in flow control mode. 0xe is the minimum value tx max blks.
3-0	P0_RX_MAX_BLKS	R/W	4h	Receive FIFO Maximum Blocks - This value is the maximum number of 1k memory blocks that may be allocated to the FIFO's logical receive queue. 0x4 is the recommended value. 0x3 is the minimum value rx max blks and 0x6 is the maximum value.

13.9 CPSW_SL Registers

Table 13-184 lists the memory-mapped registers for the CPSW_SL. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 13-184 should be considered as reserved locations and the register contents should not be modified.

Table 13-184. CPSW_SL REGISTERS

Offset	Acronym	Register Description	Section
0h	IDVER	CPGMAC_SL ID/VERSION REGISTER	Section 13.11.1
Ch	SOFT_RESET	CPGMAC_SL SOFT RESET REGISTER	Section 13.11.19
10h	RX_MAXLEN	CPGMAC_SL RX MAXIMUM LENGTH REGISTER	Section 13.9.3
14h	BOFFTEST	CPGMAC_SL BACKOFF TEST REGISTER	Section 13.9.4
18h	RX_PAUSE	CPGMAC_SL RECEIVE PAUSE TIMER REGISTER	Section 13.9.5
1Ch	TX_PAUSE	CPGMAC_SL TRANSMIT PAUSE TIMER REGISTER	Section 13.9.6
20h	EMCONTROL	CPGMAC_SL EMULATION CONTROL REGISTER	Section 13.9.7
24h	RX_PRI_MAP	CPGMAC_SL RX PKT PRIORITY TO HEADER PRIORITY MAPPING REGISTER	Section 13.9.8
4h	MACCONTROL	CPGMAC_SL MAC CONTROL REGISTER	Section 13.9.9
28h	TX_GAP	TRANSMIT INTER-PACKET GAP REGISTER	Section 13.9.10
8h	MACSTATUS	CPGMAC_SL MAC STATUS REGISTER	Section 13.9.11

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.9.1 IDVER Register (offset = 0h) [reset = 170112h]

CPGMAC_SL ID/VERSION REGISTER

IDVER is shown in Figure 13-198 and described in Table 13-211.

Figure 13-174. IDVER Register

31	30	29	28	27	26	25	24
IDENT							
R-17h							

23	22	21	20	19	18	17	16
IDENT							
R-17h							

15	14	13	12	11	10	9	8
Z					X		
R-0h					R-1h		

7	6	5	4	3	2	1	0
Y							
R-12h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-185. IDVER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	IDENT	R	17h	Rx Identification Value
15-11	Z	R	0h	Rx Z value (X.Y.Z)
10-8	X	R	1h	Rx X value (major)
7-0	Y	R	12h	Rx Y value (minor)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.9.2 SOFT_RESET Register (offset = Ch) [reset = 0h]

CPGMAC_SL SOFT RESET REGISTER

SOFT_RESET is shown in Figure 13-216 and described in Table 13-229.

Figure 13-175. SOFT_RESET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															SOFT_RESET
																															R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-186. SOFT_RESET Register Field Descriptions

Bit	Field	Type	Reset	Description
0	SOFT_RESET	R/W	0h	Software reset - Writing a one to this bit causes the CPGMAC_SL logic to be reset. After writing a one to this bit, it may be polled to determine if the reset has occurred. If a one is read, the reset has not yet occurred. If a zero is read then reset has occurred.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.9.3 RX_MAXLEN Register (offset = 10h) [reset = 5EEh]

CPGMAC_SL RX MAXIMUM LENGTH REGISTER

RX_MAXLEN is shown in Figure 13-176 and described in Table 13-187.

Figure 13-176. RX_MAXLEN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																		RX_MAXLEN													
																		R/W-5EEh													

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-187. RX_MAXLEN Register Field Descriptions

Bit	Field	Type	Reset	Description
13-0	RX_MAXLEN	R/W	5EEh	RX Maximum Frame Length - This field determines the maximum length of a received frame. The reset value is 1518 (dec). Frames with byte counts greater than rx_maxlen are long frames. Long frames with no errors are oversized frames. Long frames with CRC, code, or alignment error are jabber frames. The maximum value is 16,383.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.9.4 BOFFTEST Register (offset = 14h) [reset = 0h]

CPGMAC_SL BACKOFF TEST REGISTER

BOFFTEST is shown in Figure 13-177 and described in Table 13-188.

Figure 13-177. BOFFTEST Register

31	30	29	28	27	26	25	24
Reserved	PACEVAL					RNDNUM	
	R/W-0h					R/W-0h	

23	22	21	20	19	18	17	16
RNDNUM							
R/W-0h							

15	14	13	12	11	10	9	8
COLL_COUNT				Reserved		TX_BACKOFF	
R-0h						R-0h	

7	6	5	4	3	2	1	0
TX_BACKOFF							
R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-188. BOFFTEST Register Field Descriptions

Bit	Field	Type	Reset	Description
30-26	PACEVAL	R/W	0h	Pacing Register Current Value. A non-zero value in this field indicates that transmit pacing is active. A transmit frame collision or deferral causes paceval to loaded with decimal 31, good frame transmissions (with no collisions or deferrals) cause paceval to be decremented down to zero. When paceval is nonzero, the transmitter delays 4 IPGs between new frame transmissions after each successfully transmitted frame that had no deferrals or collisions. Transmit pacing helps reduce "capture" effects improving overall network bandwidth.
25-16	RNDNUM	R/W	0h	Backoff Random Number Generator - This field allows the Backoff Random Number Generator to be read (or written in test mode only). This field can be written only when mtest has previously been set. Reading this field returns the generator's current value. The value is reset to zero and begins counting on the clock after the deassertion of reset.
15-12	COLL_COUNT	R	0h	Collision Count - The number of collisions the current frame has experienced.
9-0	TX_BACKOFF	R	0h	Backoff Count - This field allows the current value of the backoff counter to be observed for test purposes. This field is loaded automatically according to the backoff algorithm, and is decremented by one for each slot time after the collision.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.9.5 RX_PAUSE Register (offset = 18h) [reset = 0h]

CPGMAC_SL RECEIVE PAUSE TIMER REGISTER

RX_PAUSE is shown in Figure 13-178 and described in Table 13-189.

Figure 13-178. RX_PAUSE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
15:0																Reserved															
R-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-189. RX_PAUSE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	15:0	R	0h	rx_pausetimer RX Pause Timer Value - This field allows the contents of the receive pause timer to be observed (and written in test mode). The receive pause timer is loaded with 0xFF00 when the CPGMAC_SL sends an outgoing pause frame (with pause time of 0xFFFF). The receive pause timer is decremented at slot time intervals. If the receive pause timer decrements to zero, then another outgoing pause frame will be sent and the load/decrement process will be repeated.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.9.6 TX_PAUSE Register (offset = 1Ch) [reset = 0h]

CPGMAC_SL TRANSMIT PAUSE TIMER REGISTER

TX_PAUSE is shown in Figure 13-179 and described in Table 13-190.

Figure 13-179. TX_PAUSE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
15:0																Reserved															
R-0h																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-190. TX_PAUSE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	15:0	R	0h	tx_pausetimer TX Pause Timer Value - This field allows the contents of the transmit pause timer to be observed (and written in test mode). The transmit pause timer is loaded by a received (incoming) pause frame, and then decremented, at slottime intervals, down to zero at which time CPGMAC_SL transmit frames are again enabled.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.9.7 EMCONTROL Register (offset = 20h) [reset = 0h]

CPGMAC_SL EMULATION CONTROL REGISTER

EMCONTROL is shown in Figure 13-180 and described in Table 13-191.

Figure 13-180. EMCONTROL Register

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						SOFT	FREE
						R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-191. EMCONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
1	SOFT	R/W	0h	Emulation Soft Bit
0	FREE	R/W	0h	Emulation Free Bit

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.9.8 RX_PRI_MAP Register (offset = 24h) [reset = 76543210h]

CPGMAC_SL RX PKT PRIORITY TO HEADER PRIORITY MAPPING REGISTER

RX_PRI_MAP is shown in Figure 13-181 and described in Table 13-192.

Figure 13-181. RX_PRI_MAP Register

31	30	29	28	27	26	25	24
Reserved	PRI7			Reserved	PRI6		
	R/W-7h				R/W-6h		

23	22	21	20	19	18	17	16
Reserved	PRI5			Reserved	PRI4		
	R/W-5h				R/W-4h		

15	14	13	12	11	10	9	8
Reserved	PRI3			Reserved	PRI2		
	R/W-3h				R/W-2h		

7	6	5	4	3	2	1	0
Reserved	PRI1			Reserved	PRI0		
	R/W-1h				R/W-0h		

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-192. RX_PRI_MAP Register Field Descriptions

Bit	Field	Type	Reset	Description
30-28	PRI7	R/W	7h	Priority 7 - A packet priority of 0x7 is mapped (changed) to this value.
26-24	PRI6	R/W	6h	Priority 6 - A packet priority of 0x6 is mapped (changed) to this value.
22-20	PRI5	R/W	5h	Priority 5 - A packet priority of 0x5 is mapped (changed) to this value.
18-16	PRI4	R/W	4h	Priority 4 - A packet priority of 0x4 is mapped (changed) to this value.
14-12	PRI3	R/W	3h	Priority 3 - A packet priority of 0x3 is mapped (changed) to this value.
10-8	PRI2	R/W	2h	Priority 2 - A packet priority of 0x2 is mapped (changed) to this value.
6-4	PRI1	R/W	1h	Priority 1 - A packet priority of 0x1 is mapped (changed) to this value.
2-0	PRI0	R/W	0h	Priority 0 - A packet priority of 0x0 is mapped (changed) to this value.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.9.9 MACCONTROL Register (offset = 4h) [reset = 0h]

CPGMAC_SL MAC CONTROL REGISTER

MACCONTROL is shown in Figure 13-182 and described in Table 13-193.

Figure 13-182. MACCONTROL Register

31	30	29	28	27	26	25	24
Reserved							RX_CMF_EN
							R/W-0h

23	22	21	20	19	18	17	16
RX_CSF_EN	RX_CEF_EN	TX_SHORT_GAP_LIM_EN	Reserved		EXT_EN	GIG_FORCE	IFCTL_B
R/W-0h	R/W-0h	R/W-0h			R/W-0h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
IFCTL_A	Reserved			CMD_IDLE	TX_SHORT_GAP_EN	Reserved	
R/W-0h				R/W-0h	R/W-0h		

7	6	5	4	3	2	1	0
GIG	TX_PACE	GMII_EN	TX_FLOW_EN	RX_FLOW_EN	MTEST	LOOPBACK	FULLDUPLEX
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-193. MACCONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
24	RX_CMF_EN	R/W	0h	RX Copy MAC Control Frames Enable - Enables MAC control frames to be transferred to memory. MAC control frames are normally acted upon (if enabled), but not copied to memory. MAC control frames that are pause frames will be acted upon if enabled in the MacControl register, regardless of the value of rx_cmf_en. Frames transferred to memory due to rx_cmf_en will have the control bit set in their EOP buffer descriptor. 0 - MAC control frames are filtered (but acted upon if enabled). 1 - MAC control frames are transferred to memory.
23	RX_CSF_EN	R/W	0h	RX Copy Short Frames Enable - Enables frames or fragments shorter than 64 bytes to be copied to memory. Frames transferred to memory due to rx_csf_en will have the fragment or undersized bit set in their receive footer. Fragments are short frames that contain CRC/align/code errors and undersized are short frames without errors. 0 - Short frames are filtered 1 - Short frames are transferred to memory.
22	RX_CEF_EN	R/W	0h	RX Copy Error Frames Enable - Enables frames containing errors to be transferred to memory. The appropriate error bit will be set in the frame receive footer. Frames containing errors will be filtered when rx_cef _en is not set. 0 - Frames containing errors are filtered. 1 - Frames containing errors are transferred to memory.
21	TX_SHORT_GAP_LIM_EN	R/W	0h	Transmit Short Gap Limit Enable When set this bit limits the number of short gap packets transmitted to 100ppm. Each time a short gap packet is sent, a counter is loaded with 10,000 and decremented on each wireside clock. Another short gap packet will not be sent out until the counter decrements to zero. This mode is included to preclude the host from filling up the FIFO and sending every packet out with short gap which would violate the maximum number of packets per second allowed.
18	EXT_EN	R/W	0h	Control Enable - Enables the fullduplex and gigabit mode to be selected from the FULLDUPLEX_IN and GIG_IN input signals and not from the fullduplex and gig bits in this register. The FULLDUPLEX_MODE bit reflects the actual fullduplex mode selected.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 13-193. MACCONTROL Register Field Descriptions (continued)

Bit	Field	Type	Reset	Description
17	GIG_FORCE	R/W	0h	Gigabit Mode Force - This bit is used to force the CPGMAC_SL into gigabit mode if the input GMII_MTCLK has been stopped by the PHY.
16	IFCTL_B	R/W	0h	Interface Control B - Intended as a general purpose output bit to be used to control external gaskets associated with the GMII (GMII to RGMII etc).
15	IFCTL_A	R/W	0h	Interface Control A - Intended as a general purpose output bit to be used to control external gaskets associated with the GMII (GMII to RGMII etc).
11	CMD_IDLE	R/W	0h	Command Idle 0 - Idle not commanded 1 - Idle Commanded (read idle in MacStatus)
10	TX_SHORT_GAP_EN	R/W	0h	Transmit Short Gap Enable 0 - Transmit with a short IPG is disabled 1 - Transmit with a short IPG (when TX_SHORT_GAP input is asserted) is enabled.
7	GIG	R/W	0h	Gigabit Mode - 0 - 10/100 mode 1 - Gigabit mode (full duplex only) The GIG_OUT output is the value of this bit.
6	TX_PACE	R/W	0h	Transmit Pacing Enable 0 - Transmit Pacing Disabled 1 - Transmit Pacing Enabled
5	GMII_EN	R/W	0h	GMII Enable - 0 - GMII RX and TX held in reset. 1 - GMII RX and TX released from reset.
4	TX_FLOW_EN	R/W	0h	Transmit Flow Control Enable - Determines if incoming pause frames are acted upon in full-duplex mode. Incoming pause frames are not acted upon in half-duplex mode regardless of this bit setting. The RX_MBP_Enable bits determine whether or not received pause frames are transferred to memory. 0 - Transmit Flow Control Disabled. Full-duplex mode - Incoming pause frames are not acted upon. 1 - Transmit Flow Control Enabled . Full-duplex mode - Incoming pause frames are acted upon.
3	RX_FLOW_EN	R/W	0h	Receive Flow Control Enable - 0 - Receive Flow Control Disabled Half-duplex mode - No flow control generated collisions are sent. Full-duplex mode - No outgoing pause frames are sent. 1 - Receive Flow Control Enabled Half-duplex mode - Collisions are initiated when receive flow control is triggered. Full-duplex mode - Outgoing pause frames are sent when receive flow control is triggered.
2	MTEST	R/W	0h	Manufacturing Test mode - This bit must be set to allow writes to the Backoff_Test and PauseTimer registers.
1	LOOPBACK	R/W	0h	Loop Back Mode - Loopback mode forces internal fullduplex mode regardless of whether the fullduplex bit is set or not. The loopback bit should be changed only when GMII_en is deasserted. 0 - Not looped back 1 - Loop Back Mode enabled
0	FULLDUPLEX	R/W	0h	Full Duplex mode - Gigabit mode forces fullduplex mode regardless of whether the fullduplex bit is set or not. The FULLDUPLEX_OUT output is the value of this register bit 0 - half duplex mode 1 - full duplex mode

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.9.10 TX_GAP Register (offset = 28h) [reset = Ch]

TRANSMIT INTER-PACKET GAP REGISTER

TX_GAP is shown in Figure 13-183 and described in Table 13-194.

Figure 13-183. TX_GAP Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																							TX_GAP								
																							R/W-Ch								

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-194. TX_GAP Register Field Descriptions

Bit	Field	Type	Reset	Description
8-0	TX_GAP	R/W	Ch	Transmit Inter-Packet Gap

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.9.11 MACSTATUS Register (offset = 8h) [reset = 80000000h]

CPGMAC_SL MAC STATUS REGISTER

MACSTATUS is shown in Figure 13-184 and described in Table 13-195.

Figure 13-184. MACSTATUS Register

31	30	29	28	27	26	25	24
IDLE	Reserved						
R-1h							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved			EXT_GIG	EXT_FULLDUPLEX	Reserved	RX_FLOW_ACT	TX_FLOW_ACT
			R-0h	R-0h		R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-195. MACSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31	IDLE	R	1h	CPGMAC_SL IDLE - The CPGMAC_SL is in the idle state (valid after an idle command) 0 - The CPGMAC_SL is not in the idle state. 1 - The CPGMAC_SL is in the idle state.
4	EXT_GIG	R	0h	External GIG - This is the value of the EXT_GIG input bit.
3	EXT_FULLDUPLEX	R	0h	External Fullduplex - This is the value of the EXT_FULLDUPLEX input bit.
1	RX_FLOW_ACT	R	0h	Receive Flow Control Active - When asserted, indicates that receive flow control is enabled and triggered.
0	TX_FLOW_ACT	R	0h	Transmit Flow Control Active - When asserted, this bit indicates that the pause time period is being observed for a received pause frame. No new transmissions will begin while this bit is asserted except for the transmission of pause frames. Any transmission in progress when this bit is asserted will complete.

13.10 CPSW_SS Registers

Table 13-196 lists the memory-mapped registers for the CPSW_SS. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 13-196 should be considered as reserved locations and the register contents should not be modified.

Table 13-196. CPSW_SS REGISTERS

Offset	Acronym	Register Description	Section
0h	ID_VER	ID VERSION REGISTER	Section 13.10.1
Ch	STAT_PORT_EN	STATISTICS PORT ENABLE REGISTER	Section 13.10.2
10h	PTYPE	TRANSMIT PRIORITY TYPE REGISTER	Section 13.10.3
14h	SOFT_IDLE	SOFTWARE IDLE	Section 13.10.4
18h	THRU_RATE	THROUGHPUT RATE	Section 13.10.5
1Ch	GAP_THRESH	CPGMAC_SL SHORT GAP THRESHOLD	Section 13.10.6
20h	TX_START_WDS	TRANSMIT START WORDS	Section 13.10.7
24h	FLOW_CONTROL	FLOW CONTROL	Section 13.10.8
4h	CONTROL	SWITCH CONTROL REGISTER	Section 13.11.30
28h	VLAN_LTYPE	LTYPE1 AND LTYPE 2 REGISTER	Section 13.10.10

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 13-196. CPSW_SS REGISTERS (continued)

Offset	Acronym	Register Description	Section
2Ch	TS_LTYPE	VLAN_LTYPE1 AND VLAN_LTYPE2 REGISTER	Section 13.10.11
30h	DLR_LTYPE	DLR LTYPE REGISTER	Section 13.10.12
8h	SOFT_RESET	SOFT RESET REGISTER	Section 13.11.19

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.10.1 ID_VER Register (offset = 0h) [reset = 190112h]

ID VERSION REGISTER

ID_VER is shown in Figure 13-185 and described in Table 13-197.

Figure 13-185. ID_VER Register

31	30	29	28	27	26	25	24
CPSW_3G_IDENT							
R-19h							

23	22	21	20	19	18	17	16
CPSW_3G_IDENT							
R-19h							

15	14	13	12	11	10	9	8
CPSW_3G_RTL_VER					CPSW_3G_MAJ_VER		
R-0h					R-1h		

7	6	5	4	3	2	1	0
CPSW_3G_MINOR_VER							
R-12h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-197. ID_VER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	CPSW_3G_IDENT	R	19h	3G Identification Value
15-11	CPSW_3G_RTL_VER	R	0h	3G RTL Version Value
10-8	CPSW_3G_MAJ_VER	R	1h	3G Major Version Value
7-0	CPSW_3G_MINOR_VER	R	12h	3G Minor Version Value

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.10.2 STAT_PORT_EN Register (offset = Ch) [reset = 0h]

STATISTICS PORT ENABLE REGISTER

STAT_PORT_EN is shown in Figure 13-186 and described in Table 13-198.

Figure 13-186. STAT_PORT_EN Register

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					P2_STAT_EN	P1_STAT_EN	P0_STAT_EN
					R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-198. STAT_PORT_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
2	P2_STAT_EN	R/W	0h	Port 2 (GMII2 and Port 2 FIFO) Statistics Enable 0 - Port 2 statistics are not enabled. 1 - Port 2 statistics are enabled.
1	P1_STAT_EN	R/W	0h	Port 1 (GMII1 and Port 1 FIFO) Statistics Enable 0 - Port 1 statistics are not enabled. 1 - Port 1 statistics are enabled.
0	P0_STAT_EN	R/W	0h	Port 0 Statistics Enable 0 - Port 0 statistics are not enabled 1 - Port 0 statistics are enabled. FIFO overruns (SOFOVERRUNS) are the only port 0 statistics that are enabled to be kept.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.10.3 PTYPE Register (offset = 10h) [reset = 0h]

TRANSMIT PRIORITY TYPE REGISTER

PTYPE is shown in Figure 13-187 and described in Table 13-199.

Figure 13-187. PTYPE Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved		P2_PRI3_SHAPE_EN	P2_PRI2_SHAPE_EN	P2_PRI1_SHAPE_EN	P1_PRI3_SHAPE_EN	P1_PRI2_SHAPE_EN	P1_PRI1_SHAPE_EN
		R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

15	14	13	12	11	10	9	8
Reserved					P2_PTYPE_ESC	P1_PTYPE_ESC	P0_PTYPE_ESC
					R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
Reserved			ESC_PRI_LD_VAL				
			R/W-0h				

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-199. PTYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
21	P2_PRI3_SHAPE_EN	R/W	0h	Port 2 Queue Priority 3 Transmit Shape Enable - If there is only one shaping queue then it must be priority 3.
20	P2_PRI2_SHAPE_EN	R/W	0h	Port 2 Queue Priority 2 Transmit Shape Enable - If there are two shaping queues then they must be priorities 3 and 2.
19	P2_PRI1_SHAPE_EN	R/W	0h	Port 2 Queue Priority 1 Transmit Shape Enable - If there are three shaping queues all three bits should be set.
18	P1_PRI3_SHAPE_EN	R/W	0h	Port 1 Queue Priority 3 Transmit Shape Enable - If there is only one shaping queue then it must be priority 3.
17	P1_PRI2_SHAPE_EN	R/W	0h	Port 1 Queue Priority 2 Transmit Shape Enable- If there are two shaping queues then they must be priorities 3 and 2.
16	P1_PRI1_SHAPE_EN	R/W	0h	Port 1 Queue Priority 1 Transmit Shape Enable- If there are three shaping queues all three bits should be set.
10	P2_PTYPE_ESC	R/W	0h	Port 2 Priority Type Escalate - 0 - Port 2 priority type fixed 1 - Port 2 priority type escalate Escalate should not be used with queue shaping.
9	P1_PTYPE_ESC	R/W	0h	Port 1 Priority Type Escalate - 0 - Port 1 priority type fixed 1 - Port 1 priority type escalate Escalate should not be used with queue shaping.
8	P0_PTYPE_ESC	R/W	0h	Port 0 Priority Type Escalate - 0 - Port 0 priority type fixed 1 - Port 0 priority type escalate Escalate should not be used with queue shaping.
4-0	ESC_PRI_LD_VAL	R/W	0h	Escalate Priority Load Value When a port is in escalate priority, this is the number of higher priority packets sent before the next lower priority is allowed to send a packet. Escalate priority allows lower priority packets to be sent at a fixed rate relative to the next higher priority.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.10.4 SOFT_IDLE Register (offset = 14h) [reset = 0h]

SOFTWARE IDLE

SOFT_IDLE is shown in Figure 13-188 and described in Table 13-200.

Figure 13-188. SOFT_IDLE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															SOFT_IDLE
																															R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-200. SOFT_IDLE Register Field Descriptions

Bit	Field	Type	Reset	Description
0	SOFT_IDLE	R/W	0h	Software Idle - Setting this bit causes the switch fabric to stop forwarding packets at the next start of packet.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.10.5 THRU_RATE Register (offset = 18h) [reset = 3003h]

THROUGHPUT RATE

THRU_RATE is shown in Figure 13-189 and described in Table 13-201.

Figure 13-189. THRU_RATE Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
SL_RX_THRU_RATE				Reserved			
R/W-3h							

7	6	5	4	3	2	1	0
Reserved				CPDMA_THRU_RATE			
				R/W-3h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-201. THRU_RATE Register Field Descriptions

Bit	Field	Type	Reset	Description
15-12	SL_RX_THRU_RATE	R/W	3h	CPGMAC_SL Switch FIFO receive through rate. This register value is the maximum throughput of the ethernet ports to the crossbar SCR. The default is one 8-byte word for every 3 CLK periods maximum.
3-0	CPDMA_THRU_RATE	R/W	3h	CPDMA Switch FIFO receive through rate. This register value is the maximum throughput of the CPDMA host port to the crossbar SCR. The default is one 8-byte word for every 3 CLK periods maximum.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.10.6 GAP_THRESH Register (offset = 1Ch) [reset = Bh]

CPGMAC_SL SHORT GAP THRESHOLD

GAP_THRESH is shown in Figure 13-190 and described in Table 13-202.

Figure 13-190. GAP_THRESH Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																											GAP_THRESH				
																											R/W-Bh				

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-202. GAP_THRESH Register Field Descriptions

Bit	Field	Type	Reset	Description
4-0	GAP_THRESH	R/W	Bh	CPGMAC_SL Short Gap Threshold - This is the CPGMAC_SL associated FIFO transmit block usage value for triggering TX_SHORT_GAP.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.10.7 TX_START_WDS Register (offset = 20h) [reset = 20h]

TRANSMIT START WORDS

TX_START_WDS is shown in Figure 13-191 and described in Table 13-203.

Figure 13-191. TX_START_WDS Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																					TX_START_WDS										
																					R/W-20h										

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-203. TX_START_WDS Register Field Descriptions

Bit	Field	Type	Reset	Description
10-0	TX_START_WDS	R/W	20h	FIFO Packet Transmit (egress) Start Words. This value is the number of required packet words in the transmit FIFO before the packet egress will begin. This value is non-zero to preclude underrun. Decimal 32 is the recommended value. It should not be increased unnecessairly to prevent adding to the switch latency.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.10.8 FLOW_CONTROL Register (offset = 24h) [reset = 1h]

FLOW CONTROL

FLOW_CONTROL is shown in Figure 13-192 and described in Table 13-204.

Figure 13-192. FLOW_CONTROL Register

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					P2_FLOW_EN	P1_FLOW_EN	P0_FLOW_EN
					R/W-0h	R/W-0h	R/W-1h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-204. FLOW_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
2	P2_FLOW_EN	R/W	0h	Port 2 Receive flow control enable
1	P1_FLOW_EN	R/W	0h	Port 1 Receive flow control enable
0	P0_FLOW_EN	R/W	1h	Port 0 Receive flow control enable

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.10.9 CONTROL Register (offset = 4h) [reset = 0h]

SWITCH CONTROL REGISTER

CONTROL is shown in Figure 13-227 and described in Table 13-240.

Figure 13-193. CONTROL Register

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				DLR_EN	RX_VLAN_ENCAP	VLAN_AWARE	FIFO_LOOPBACK
				R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-205. CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
3	DLR_EN	R/W	0h	DLR enable 0 - DLR is disabled. DLR packets will not be moved to queue priority 3 and will not be separated out onto dlr_cpdma_ch. 1 - DLR is disabled. DLR packets be moved to destination port transmit queue priority 3 and will be separated out onto dlr_cpdma_ch when packet is to egress on port 0.
2	RX_VLAN_ENCAP	R/W	0h	Port 0 VLAN Encapsulation (egress): 0 - Port 2 receive packets (from 3G) are not VLAN encapsulated. 1 - Port 2 receive packets (from 3G) are VLAN encapsulated.
1	VLAN_AWARE	R/W	0h	VLAN Aware Mode: 0 - 3G is in the VLAN unaware mode. 1 - 3G is in the VLAN aware mode.
0	FIFO_LOOPBACK	R/W	0h	FIFO Loopback Mode 0 - Loopback is disabled 1 - FIFO Loopback mode enabled. Each packet received is turned around and sent out on the same port's transmit path. Port 2 receive is fixed on channel zero. The RXSOFOVERRUN statistic will increment for every packet sent in FIFO loopback mode.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.10.10 VLAN_LTYPE Register (offset = 28h) [reset = 81008100h]

LTYPE1 AND LTYPE 2 REGISTER

VLAN_LTYPE is shown in Figure 13-194 and described in Table 13-206.

Figure 13-194. VLAN_LTYPE Register

31	30	29	28	27	26	25	24
VLAN_LTYPE2							
R/W-8100h							

23	22	21	20	19	18	17	16
VLAN_LTYPE2							
R/W-8100h							

15	14	13	12	11	10	9	8
VLAN_LTYPE1							
R/W-8100h							

7	6	5	4	3	2	1	0
VLAN_LTYPE1							
R/W-8100h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-206. VLAN_LTYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
31-16	VLAN_LTYPE2	R/W	8100h	Time Sync VLAN LTYPE2 This VLAN LTYPE value is used for tx and rx. This is the inner VLAN if both are present.
15-0	VLAN_LTYPE1	R/W	8100h	Time Sync VLAN LTYPE1 This VLAN LTYPE value is used for tx and rx. This is the outer VLAN if both are present.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.10.11 TS_LTYPE Register (offset = 2Ch) [reset = 0h]

VLAN_LTYPE1 AND VLAN_LTYPE2 REGISTER

TS_LTYPE is shown in Figure 13-195 and described in Table 13-207.

Figure 13-195. TS_LTYPE Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved		TS_LTYPE2					
		R/W-0h					

15	14	13	12	11	10	9	8
TS_LTYPE1							
R/W-0h							

7	6	5	4	3	2	1	0
TS_LTYPE1							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-207. TS_LTYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
21-16	TS_LTYPE2	R/W	0h	Time Sync LTYPE2 This is an Ethertype value to match for tx and rx time sync packets.
15-0	TS_LTYPE1	R/W	0h	Time Sync LTYPE1 This is an ethertype value to match for tx and rx time sync packets.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.10.12 DLR_LTYPE Register (offset = 30h) [reset = 80E1h]

DLR LTYPE REGISTER

DLR_LTYPE is shown in Figure 13-196 and described in Table 13-208.

Figure 13-196. DLR_LTYPE Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																DLR_LTYPE															
																							R/W-80E1h								

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-208. DLR_LTYPE Register Field Descriptions

Bit	Field	Type	Reset	Description
15-0	DLR_LTYPE	R/W	80E1h	DLR LTYPE

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.10.13 SOFT_RESET Register (offset = 8h) [reset = 0h]

SOFT RESET REGISTER

SOFT_RESET is shown in Figure 13-216 and described in Table 13-229.

Figure 13-197. SOFT_RESET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															SOFT_RESET
																															R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-209. SOFT_RESET Register Field Descriptions

Bit	Field	Type	Reset	Description
0	SOFT_RESET	R/W	0h	Software reset - Writing a one to this bit causes the 3G logic to be reset. After writing a one to this bit, it may be polled to determine if the reset has occurred. If a one is read, the reset has not yet occurred. If a zero is read then reset has occurred.

13.11 CPSW_WR Registers

Table 13-210 lists the memory-mapped registers for the CPSW_WR. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 13-210 should be considered as reserved locations and the register contents should not be modified.

Table 13-210. CPSW_WR REGISTERS

Offset	Acronym	Register Description	Section
0h	IDVER	SUBSYSTEM ID VERSION REGISTER	Section 13.11.1
64h	C2_RX_STAT	SUBSYSTEM CORE 2 RECEIVE MASKED INTERRUPT STATUS REGISTER	Section 13.11.2
68h	C2_TX_STAT	SUBSYSTEM CORE 2 TRANSMIT MASKED INTERRUPT STATUS REGISTER	Section 13.11.3
6Ch	C2_MISC_STAT	SUBSYSTEM CORE 2 MISC MASKED INTERRUPT STATUS REGISTER	Section 13.11.4
70h	C0_RX_IMAX	SUBSYSTEM CORE 0 RECEIVE INTERRUPTS PER MILLISECOND	Section 13.11.5
74h	C0_TX_IMAX	SUBSYSTEM CORE 0 TRANSMIT INTERRUPTS PER MILLISECOND	Section 13.11.6
Ch	INT_CONTROL	SUBSYSTEM INTERRUPT CONTROL	Section 13.11.7
78h	C1_RX_IMAX	SUBSYSTEM CORE 1 RECEIVE INTERRUPTS PER MILLISECOND	Section 13.11.8
7Ch	C1_TX_IMAX	SUBSYSTEM CORE 1 TRANSMIT INTERRUPTS PER MILLISECOND	Section 13.11.9
80h	C2_RX_IMAX	SUBSYSTEM CORE 2 RECEIVE INTERRUPTS PER MILLISECOND	Section 13.11.10
84h	C2_TX_IMAX	SUBSYSTEM CORE 2 TRANSMIT INTERRUPTS PER MILLISECOND	Section 13.11.11
88h	RGMII_CTL	RGMII CONTROL SIGNAL REGISTER	Section 13.11.12
10h	C0_RX_THRESH_EN	SUBSYSTEM CORE 0 RECEIVE THRESHOLD INT ENABLE REGISTER	Section 13.11.13

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 13-210. CPSW_WR REGISTERS (continued)

Offset	Acronym	Register Description	Section
14h	C0_RX_EN	SUBSYSTEM CORE 0 RECEIVE INTERRUPT ENABLE REGISTER	Section 13.11.14
18h	C0_TX_EN	SUBSYSTEM CORE 0 TRANSMIT INTERRUPT ENABLE REGISTER	Section 13.11.15
1Ch	C0_MISC_EN	SUBSYSTEM CORE 0 MISC INTERRUPT ENABLE REGISTER	Section 13.11.16
20h	C1_RX_THRESH_EN	SUBSYSTEM CORE 1 RECEIVE THRESHOLD INT ENABLE REGISTER	Section 13.11.17
24h	C1_RX_EN	SUBSYSTEM CORE 1 RECEIVE INTERRUPT ENABLE REGISTER	Section 13.11.18
4h	SOFT_RESET	SUBSYSTEM SOFT RESET REGISTER	Section 13.11.19
28h	C1_TX_EN	SUBSYSTEM CORE 1 TRANSMIT INTERRUPT ENABLE REGISTER	Section 13.11.20
2Ch	C1_MISC_EN	SUBSYSTEM CORE 1 MISC INTERRUPT ENABLE REGISTER	Section 13.11.21
30h	C2_RX_THRESH_EN	SUBSYSTEM CORE 2 RECEIVE THRESHOLD INT ENABLE REGISTER	Section 13.11.22
34h	C2_RX_EN	SUBSYSTEM CORE 2 RECEIVE INTERRUPT ENABLE REGISTER	Section 13.11.23
38h	C2_TX_EN	SUBSYSTEM CORE 2 TRANSMIT INTERRUPT ENABLE REGISTER	Section 13.11.24
3Ch	C2_MISC_EN	SUBSYSTEM CORE 2 MISC INTERRUPT ENABLE REGISTER	Section 13.11.25
40h	C0_RX_THRESH_STAT	SUBSYSTEM CORE 0 RX THRESHOLD MASKED INT STATUS REGISTER	Section 13.11.26
44h	C0_RX_STAT	SUBSYSTEM CORE 0 RX INTERRUPT MASKED INT STATUS REGISTER	Section 13.11.27
48h	C0_TX_STAT	SUBSYSTEM CORE 0 TX INTERRUPT MASKED INT STATUS REGISTER	Section 13.11.28
4Ch	C0_MISC_STAT	SUBSYSTEM CORE 0 MISC INTERRUPT MASKED INT STATUS REGISTER	Section 13.11.29
8h	CONTROL	SUBSYSTEM CONTROL REGISTER	Section 13.11.30
50h	C1_RX_THRESH_STAT	SUBSYSTEM CORE 1 RX THRESHOLD MASKED INT STATUS REGISTER	Section 13.11.31
54h	C1_RX_STAT	SUBSYSTEM CORE 1 RECEIVE MASKED INTERRUPT STATUS REGISTER	Section 13.11.32
58h	C1_TX_STAT	SUBSYSTEM CORE 1 TRANSMIT MASKED INTERRUPT STATUS REGISTER	Section 13.11.33
5Ch	C1_MISC_STAT	SUBSYSTEM CORE 1 MISC MASKED INTERRUPT STATUS REGISTER	Section 13.11.34
60h	C2_RX_THRESH_STAT	SUBSYSTEM CORE 2 RX THRESHOLD MASKED INT STATUS REGISTER	Section 13.11.35

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.1 IDVER Register (offset = 0h) [reset = EEB0100h]

SUBSYSTEM ID VERSION REGISTER

IDVER is shown in Figure 13-198 and described in Table 13-211.

Figure 13-198. IDVER Register

31	30	29	28	27	26	25	24
SCHEME							
R-1h							

23	22	21	20	19	18	17	16
SCHEME				Reserved			
R-1h							

15	14	13	12	11	10	9	8
RTL					MAJOR		
R-0h					R-1h		

7	6	5	4	3	2	1	0
CUSTOM		MINOR					
R-0h		R-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-211. IDVER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-20	SCHEME	R	1h	Scheme value
27-16	FUNCTION	R	EDBh	function value
15-11	RTL	R	0h	rtl version
10-8	MAJOR	R	1h	major version
7-6	CUSTOM	R	0h	custom version
5-0	MINOR	R	0h	minor version

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.2 C2_RX_STAT Register (offset = 64h) [reset = 0h]

SUBSYSTEM CORE 2 RECEIVE MASKED INTERRUPT STATUS REGISTER

C2_RX_STAT is shown in Figure 13-199 and described in Table 13-212.

Figure 13-199. C2_RX_STAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								C2_RX_STAT							
																								R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-212. C2_RX_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	C2_RX_STAT	R	0h	Core 2 Receive Masked Interrupt Status - Each bit in this register corresponds to the bit in the Rx interrupt that is enabled and generating an interrupt on C2_RX_PULSE.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.3 C2_TX_STAT Register (offset = 68h) [reset = 0h]

SUBSYSTEM CORE 2 TRANSMIT MASKED INTERRUPT STATUS REGISTER

C2_TX_STAT is shown in Figure 13-200 and described in Table 13-213.

Figure 13-200. C2_TX_STAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								C2_TX_STAT							
																								R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-213. C2_TX_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	C2_TX_STAT	R	0h	Core 2 Transmit Masked Interrupt Status - Each bit in this register corresponds to the bit in the Tx interrupt that is enabled and generating an interrupt on C2_TX_PULSE.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.4 C2_MISC_STAT Register (offset = 6Ch) [reset = 0h]

SUBSYSTEM CORE 2 MISC MASKED INTERRUPT STATUS REGISTER

C2_MISC_STAT is shown in Figure 13-201 and described in Table 13-214.

Figure 13-201. C2_MISC_STAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																											C2_MISC_STAT				
																											R-0h				

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-214. C2_MISC_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
4-0	C2_MISC_STAT	R	0h	Core 2 Misc Masked Interrupt Status - Each bit in this register corresponds to the miscellaneous interrupt (evnt_pend, stat_pend, host_pend, mdio_linkint, mdio_userint) that is enabled and generating an interrupt on C2_MISC_PULSE .

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.5 C0_RX_IMAX Register (offset = 70h) [reset = 0h]

SUBSYSTEM CORE 0 RECEIVE INTERRUPTS PER MILLISECOND

C0_RX_IMAX is shown in Figure 13-202 and described in Table 13-215.

Figure 13-202. C0_RX_IMAX Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																										C0_RX_IMAX					
																										R/W-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-215. C0_RX_IMAX Register Field Descriptions

Bit	Field	Type	Reset	Description
5-0	C0_RX_IMAX	R/W	0h	Core 0 Receive Interrupts per Millisecond - The maximum number of interrupts per millisecond generated on C0_RX_PULSE if pacing is enabled for this interrupt.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.6 C0_TX_IMAX Register (offset = 74h) [reset = 0h]

SUBSYSTEM CORE 0 TRANSMIT INTERRUPTS PER MILLISECOND

C0_TX_IMAX is shown in Figure 13-203 and described in Table 13-216.

Figure 13-203. C0_TX_IMAX Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																										C0_TX_IMAX					
																										R/W-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-216. C0_TX_IMAX Register Field Descriptions

Bit	Field	Type	Reset	Description
5-0	C0_TX_IMAX	R/W	0h	Core 0 Transmit Interrupts per Millisecond - The maximum number of interrupts per millisecond generated on C0_TX_PULSE if pacing is enabled for this interrupt.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.7 INT_CONTROL Register (offset = Ch) [reset = 0h]

SUBSYSTEM INTERRUPT CONTROL

INT_CONTROL is shown in Figure 13-204 and described in Table 13-217.

Figure 13-204. INT_CONTROL Register

31	30	29	28	27	26	25	24
INT_TEST	Reserved						
R/W-0h							

23	22	21	20	19	18	17	16
Reserved		INT_PACE_EN					
		R/W-0h					

15	14	13	12	11	10	9	8
Reserved				INT_PRESCALE			
				R-0h			

7	6	5	4	3	2	1	0
INT_PRESCALE							
R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-217. INT_CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
31	INT_TEST	R/W	0h	Interrupt Test - Test bit to the interrupt pacing blocks
21-16	INT_PACE_EN	R/W	0h	Interrupt Pacing Enable Bus int_pace_en[0] - Enables C0_Rx_Pulse Pacing (0 is pacing bypass) int_pace_en[1] - Enables C0_Tx_Pulse Pacing (0 is pacing bypass) int_pace_en[2] - Enables C1_Rx_Pulse Pacing (0 is pacing bypass) int_pace_en[3] - Enables C1_Tx_Pulse Pacing (0 is pacing bypass) int_pace_en[4] - Enables C2_Rx_Pulse Pacing (0 is pacing bypass) int_pace_en[5] - Enables C2_Tx_Pulse Pacing (0 is pacing bypass)
11-0	INT_PRESCALE	R	0h	Interrupt Counter Prescaler - The number of MAIN_CLK periods in 4us.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.8 C1_RX_IMAX Register (offset = 78h) [reset = 0h]

SUBSYSTEM CORE 1 RECEIVE INTERRUPTS PER MILLISECOND

C1_RX_IMAX is shown in Figure 13-205 and described in Table 13-218.

Figure 13-205. C1_RX_IMAX Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																										C1_RX_IMAX					
																										R/W-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-218. C1_RX_IMAX Register Field Descriptions

Bit	Field	Type	Reset	Description
5-0	C1_RX_IMAX	R/W	0h	Core 1 Receive Interrupts per Millisecond - The maximum number of interrupts per millisecond generated on C1_RX_PULSE if pacing is enabled for this interrupt.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.9 C1_TX_IMAX Register (offset = 7Ch) [reset = 0h]

SUBSYSTEM CORE 1 TRANSMIT INTERRUPTS PER MILLISECOND

C1_TX_IMAX is shown in Figure 13-206 and described in Table 13-219.

Figure 13-206. C1_TX_IMAX Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																										C1_TX_IMAX					
																										R/W-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-219. C1_TX_IMAX Register Field Descriptions

Bit	Field	Type	Reset	Description
5-0	C1_TX_IMAX	R/W	0h	Core 1 Transmit Interrupts per Millisecond - The maximum number of interrupts per millisecond generated on C1_TX_PULSE if pacing is enabled for this interrupt.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.10 C2_RX_IMAX Register (offset = 80h) [reset = 0h]

SUBSYSTEM CORE 2 RECEIVE INTERRUPTS PER MILLISECOND

C2_RX_IMAX is shown in Figure 13-207 and described in Table 13-220.

Figure 13-207. C2_RX_IMAX Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																										C2_RX_IMAX					
																										R/W-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-220. C2_RX_IMAX Register Field Descriptions

Bit	Field	Type	Reset	Description
5-0	C2_RX_IMAX	R/W	0h	Core 2 Receive Interrupts per Millisecond - The maximum number of interrupts per millisecond generated on C2_RX_PULSE if pacing is enabled for this interrupt.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.11 C2_TX_IMAX Register (offset = 84h) [reset = 0h]

SUBSYSTEM CORE 2 TRANSMIT INTERRUPTS PER MILLISECOND

C2_TX_IMAX is shown in Figure 13-208 and described in Table 13-221.

Figure 13-208. C2_TX_IMAX Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																										C2_TX_IMAX					
																										R/W-0h					

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-221. C2_TX_IMAX Register Field Descriptions

Bit	Field	Type	Reset	Description
5-0	C2_TX_IMAX	R/W	0h	Core 2 Transmit Interrupts per Millisecond - The maximum number of interrupts per millisecond generated on C2_TX_PULSE if pacing is enabled for this interrupt.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.12 RGMII_CTL Register (offset = 88h) [reset = 0h]

RGMII CONTROL SIGNAL REGISTER

RGMII_CTL is shown in Figure 13-209 and described in Table 13-222.

Figure 13-209. RGMII_CTL Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
RGMII2_FULLDUPLEX	RGMII2_SPEED		RGMII2_LINK	RGMII1_FULLDUPLEX	RGMII1_SPEED		RGMII1_LINK
R-0h	R-0h		R-0h	R-0h	R-0h		R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-222. RGMII_CTL Register Field Descriptions

Bit	Field	Type	Reset	Description
7	RGMII2_FULLDUPLEX	R	0h	RGMII 2 Fullduplex - This is the CPRGMII fullduplex output signal. 0 - Half-duplex mode 1 - Fullduplex mode
6-5	RGMII2_SPEED	R	0h	RGMII2 Speed - This is the CPRGMI speed output signal 00 - 10Mbps mode 01 - 100Mbps mode 10 - 1000Mbps (gig) mode 11 - reserved
4	RGMII2_LINK	R	0h	RGMII2 Link Indicator - This is the CPRGMII link output signal 0 - RGMII2 link is down 1 - RGMII2 link is up
3	RGMII1_FULLDUPLEX	R	0h	RGMII1 Fullduplex - This is the CPRGMII fullduplex output signal. 0 - Half-duplex mode 1 - Fullduplex mode
2-1	RGMII1_SPEED	R	0h	RGMII1 Speed - This is the CPRGMII speed output signal 00 - 10Mbps mode 01 - 100Mbps mode 10 - 1000Mbps (gig) mode 11 - reserved
0	RGMII1_LINK	R	0h	RGMII1 Link Indicator - This is the CPRGMII link output signal 0 - RGMII1 link is down

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.13 C0_RX_THRESH_EN Register (offset = 10h) [reset = 0h]

SUBSYSTEM CORE 0 RECEIVE THRESHOLD INT ENABLE REGISTER

C0_RX_THRESH_EN is shown in Figure 13-210 and described in Table 13-223.

Figure 13-210. C0_RX_THRESH_EN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								C0_RX_THRESH_EN							
																								R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-223. C0_RX_THRESH_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	C0_RX_THRESH_EN	R/W	0h	Core 0 Receive Threshold Enable - Each bit in this register corresponds to the bit in the receive threshold interrupt that is enabled to generate an interrupt on C0_RX_THRESH_PULSE.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.14 C0_RX_EN Register (offset = 14h) [reset = 0h]

SUBSYSTEM CORE 0 RECEIVE INTERRUPT ENABLE REGISTER

C0_RX_EN is shown in Figure 13-211 and described in Table 13-224.

Figure 13-211. C0_RX_EN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								C0_RX_EN							
																								R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-224. C0_RX_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	C0_RX_EN	R/W	0h	Core 0 Receive Enable - Each bit in this register corresponds to the bit in the rx interrupt that is enabled to generate an interrupt on C0_RX_PULSE.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.15 C0_TX_EN Register (offset = 18h) [reset = 0h]

SUBSYSTEM CORE 0 TRANSMIT INTERRUPT ENABLE REGISTER

C0_TX_EN is shown in Figure 13-212 and described in Table 13-225.

Figure 13-212. C0_TX_EN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								C0_TX_EN							
																								R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-225. C0_TX_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	C0_TX_EN	R/W	0h	Core 0 Transmit Enable - Each bit in this register corresponds to the bit in the tx interrupt that is enabled to generate an interrupt on C0_TX_PULSE.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.16 C0_MISC_EN Register (offset = 1Ch) [reset = 0h]

SUBSYSTEM CORE 0 MISC INTERRUPT ENABLE REGISTER

C0_MISC_EN is shown in Figure 13-213 and described in Table 13-226.

Figure 13-213. C0_MISC_EN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																											C0_MISC_EN				
																											R/W-0h				

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-226. C0_MISC_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
4-0	C0_MISC_EN	R/W	0h	Core 0 Misc Enable - Each bit in this register corresponds to the miscellaneous interrupt (evnt_pend, stat_pend, host_pend, mdio_linkint, mdio_userint) that is enabled to generate an interrupt on C0_Misc_PULSE.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.17 C1_RX_THRESH_EN Register (offset = 20h) [reset = 0h]

SUBSYSTEM CORE 1 RECEIVE THRESHOLD INT ENABLE REGISTER

C1_RX_THRESH_EN is shown in Figure 13-214 and described in Table 13-227.

Figure 13-214. C1_RX_THRESH_EN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								C1_RX_THRESH_EN							
																								R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-227. C1_RX_THRESH_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	C1_RX_THRESH_EN	R/W	0h	Core 1 Receive Threshold Enable - Each bit in this register corresponds to the bit in the receive threshold interrupt that is enabled to generate an interrupt on C1_RX_THRESH_PULSE.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.18 C1_RX_EN Register (offset = 24h) [reset = 0h]

SUBSYSTEM CORE 1 RECEIVE INTERRUPT ENABLE REGISTER

C1_RX_EN is shown in Figure 13-215 and described in Table 13-228.

Figure 13-215. C1_RX_EN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								C1_RX_EN							
																								R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-228. C1_RX_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	C1_RX_EN	R/W	0h	Core 1 Receive Enable - Each bit in this register corresponds to the bit in the rx interrupt that is enabled to generate an interrupt on C1_RX_PULSE.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.19 SOFT_RESET Register (offset = 4h) [reset = 0h]

SUBSYSTEM SOFT RESET REGISTER

SOFT_RESET is shown in Figure 13-216 and described in Table 13-229.

Figure 13-216. SOFT_RESET Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																															SOFT_RESET
																															R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-229. SOFT_RESET Register Field Descriptions

Bit	Field	Type	Reset	Description
0	SOFT_RESET	R/W	0h	Software reset - Writing a one to this bit causes the CPGMACSS_R logic to be reset (INT, REGS, CPPI). Software reset occurs on the clock following the register bit write.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.20 C1_TX_EN Register (offset = 28h) [reset = 0h]

SUBSYSTEM CORE 1 TRANSMIT INTERRUPT ENABLE REGISTER

C1_TX_EN is shown in Figure 13-217 and described in Table 13-230.

Figure 13-217. C1_TX_EN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								C1_TX_EN							
																								R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-230. C1_TX_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	C1_TX_EN	R/W	0h	Core 1 Transmit Enable - Each bit in this register corresponds to the bit in the tx interrupt that is enabled to generate an interrupt on C1_TX_PULSE.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.21 C1_MISC_EN Register (offset = 2Ch) [reset = 0h]

SUBSYSTEM CORE 1 MISC INTERRUPT ENABLE REGISTER

C1_MISC_EN is shown in Figure 13-218 and described in Table 13-231.

Figure 13-218. C1_MISC_EN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																											C1_MISC_EN				
																											R/W-0h				

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-231. C1_MISC_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
4-0	C1_MISC_EN	R/W	0h	Core 1 Misc Enable - Each bit in this register corresponds to the miscellaneous interrupt (evnt_pend, stat_pend, host_pend, mdio_linkint, mdio_userint) that is enabled to generate an interrupt on C1_Misc_PULSE.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.22 C2_RX_THRESH_EN Register (offset = 30h) [reset = 0h]

SUBSYSTEM CORE 2 RECEIVE THRESHOLD INT ENABLE REGISTER

C2_RX_THRESH_EN is shown in Figure 13-219 and described in Table 13-232.

Figure 13-219. C2_RX_THRESH_EN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								C2_RX_THRESH_EN							
																								R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-232. C2_RX_THRESH_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	C2_RX_THRESH_EN	R/W	0h	Core 2 Receive Threshold Enable - Each bit in this register corresponds to the bit in the receive threshold interrupt that is enabled to generate an interrupt on C2_RX_THRESH_PULSE.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.23 C2_RX_EN Register (offset = 34h) [reset = 0h]

SUBSYSTEM CORE 2 RECEIVE INTERRUPT ENABLE REGISTER

C2_RX_EN is shown in Figure 13-220 and described in Table 13-233.

Figure 13-220. C2_RX_EN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								C2_RX_EN							
																								R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-233. C2_RX_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	C2_RX_EN	R/W	0h	Core 2 Receive Enable - Each bit in this register corresponds to the bit in the rx interrupt that is enabled to generate an interrupt on C2_RX_PULSE.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.24 C2_TX_EN Register (offset = 38h) [reset = 0h]

SUBSYSTEM CORE 2 TRANSMIT INTERRUPT ENABLE REGISTER

C2_TX_EN is shown in Figure 13-221 and described in Table 13-234.

Figure 13-221. C2_TX_EN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								C2_TX_EN							
																								R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-234. C2_TX_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	C2_TX_EN	R/W	0h	Core 2 Transmit Enable - Each bit in this register corresponds to the bit in the tx interrupt that is enabled to generate an interrupt on C2_TX_PULSE.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.25 C2_MISC_EN Register (offset = 3Ch) [reset = 0h]

SUBSYSTEM CORE 2 MISC INTERRUPT ENABLE REGISTER

C2_MISC_EN is shown in Figure 13-222 and described in Table 13-235.

Figure 13-222. C2_MISC_EN Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																											C2_MISC_EN				
																											R/W-0h				

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-235. C2_MISC_EN Register Field Descriptions

Bit	Field	Type	Reset	Description
4-0	C2_MISC_EN	R/W	0h	Core 2 Misc Enable - Each bit in this register corresponds to the miscellaneous interrupt (evnt_pend, stat_pend, host_pend, mdio_linkint, mdio_userint) that is enabled to generate an interrupt on C2_Misc_PULSE.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.26 C0_RX_THRESH_STAT Register (offset = 40h) [reset = 0h]

SUBSYSTEM CORE 0 RX THRESHOLD MASKED INT STATUS REGISTER

C0_RX_THRESH_STAT is shown in Figure 13-223 and described in Table 13-236.

Figure 13-223. C0_RX_THRESH_STAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								C0_RX_THRESH_STAT							
																								R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-236. C0_RX_THRESH_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	C0_RX_THRESH_STAT	R	0h	Core 0 Receive Threshold Masked Interrupt Status - Each bit in this read only register corresponds to the bit in the receive threshold interrupt that is enabled and generating an interrupt on C0_RX_THRESH_PULSE.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.27 C0_RX_STAT Register (offset = 44h) [reset = 0h]

SUBSYSTEM CORE 0 RX INTERRUPT MASKED INT STATUS REGISTER

C0_RX_STAT is shown in Figure 13-224 and described in Table 13-237.

Figure 13-224. C0_RX_STAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								C0_RX_STAT							
																								R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-237. C0_RX_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	C0_RX_STAT	R	0h	Core 0 Receive Masked Interrupt Status - Each bit in this read only register corresponds to the bit in the Rx interrupt that is enabled and generating an interrupt on C0_RX_PULSE.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.28 C0_TX_STAT Register (offset = 48h) [reset = 0h]

SUBSYSTEM CORE 0 TX INTERRUPT MASKED INT STATUS REGISTER

C0_TX_STAT is shown in Figure 13-225 and described in Table 13-238.

Figure 13-225. C0_TX_STAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								C0_TX_STAT							
																								R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-238. C0_TX_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	C0_TX_STAT	R	0h	Core 0 Transmit Masked Interrupt Status - Each bit in this read only register corresponds to the bit in the Tx interrupt that is enabled and generating an interrupt on C0_TX_PULSE .

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.29 C0_MISC_STAT Register (offset = 4Ch) [reset = 0h]

SUBSYSTEM CORE 0 MISC INTERRUPT MASKED INT STATUS REGISTER

C0_MISC_STAT is shown in Figure 13-226 and described in Table 13-239.

Figure 13-226. C0_MISC_STAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																											C0_MISC_STAT				
																											R-0h				

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-239. C0_MISC_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
4-0	C0_MISC_STAT	R	0h	Core 0 Misc Masked Interrupt Status - Each bit in this register corresponds to the miscellaneous interrupt (evnt_pend, stat_pend, host_pend, mdio_linkint, mdio_userint) that is enabled and generating an interrupt on C0_MISC_PULSE .

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.30 CONTROL Register (offset = 8h) [reset = Ah]

SUBSYSTEM CONTROL REGISTER

CONTROL is shown in Figure 13-227 and described in Table 13-240.

Figure 13-227. CONTROL Register

31	30	29	28	27	26	25	24
Reserved							

23	22	21	20	19	18	17	16
Reserved							

15	14	13	12	11	10	9	8
Reserved							

7	6	5	4	3	2	1	0
Reserved				MMR_STDBYMODE		MMR_IDLEMODE	
				R/W-2h		R/W-2h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-240. CONTROL Register Field Descriptions

Bit	Field	Type	Reset	Description
3-2	MMR_STDBYMODE	R/W	2h	Standbymode MMR bits - This field is used to cause the CPSW_3GSS_R to go to standby mode (software standby). Smart standby is not supported.
1-0	MMR_IDLEMODE	R/W	2h	Idlemode MMR bits - This field is used to cause the CPSW_3GSS_R to go to idle mode (software idle). Smart idle is not supported.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.31 C1_RX_THRESH_STAT Register (offset = 50h) [reset = 0h]

SUBSYSTEM CORE 1 RX THRESHOLD MASKED INT STATUS REGISTER

C1_RX_THRESH_STAT is shown in Figure 13-228 and described in Table 13-241.

Figure 13-228. C1_RX_THRESH_STAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								C1_RX_THRESH_STAT							
																								R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-241. C1_RX_THRESH_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	C1_RX_THRESH_STAT	R	0h	Core 1 Receive Threshold Masked Interrupt Status - Each bit in this register corresponds to the bit in the receive threshold interrupt that is enabled and generating an interrupt on C1_RX_THRESH_PULSE.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.32 C1_RX_STAT Register (offset = 54h) [reset = 0h]

SUBSYSTEM CORE 1 RECEIVE MASKED INTERRUPT STATUS REGISTER

C1_RX_STAT is shown in Figure 13-229 and described in Table 13-242.

Figure 13-229. C1_RX_STAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								C1_RX_STAT							
																								R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-242. C1_RX_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	C1_RX_STAT	R	0h	Core 1 Receive Masked Interrupt Status - Each bit in this register corresponds to the bit in the Rx interrupt that is enabled and generating an interrupt on C1_RX_PULSE.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.33 C1_TX_STAT Register (offset = 58h) [reset = 0h]

SUBSYSTEM CORE 1 TRANSMIT MASKED INTERRUPT STATUS REGISTER

C1_TX_STAT is shown in Figure 13-230 and described in Table 13-243.

Figure 13-230. C1_TX_STAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								C1_TX_STAT							
																								R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-243. C1_TX_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	C1_TX_STAT	R	0h	Core 1 Transmit Masked Interrupt Status - Each bit in this register corresponds to the bit in the Tx interrupt that is enabled and generating an interrupt on C1_TX_PULSE.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.34 C1_MISC_STAT Register (offset = 5Ch) [reset = 0h]

SUBSYSTEM CORE 1 MISC MASKED INTERRUPT STATUS REGISTER

C1_MISC_STAT is shown in Figure 13-231 and described in Table 13-244.

Figure 13-231. C1_MISC_STAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																											C1_MISC_STAT				
																											R-0h				

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 13-244. C1_MISC_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
4-0	C1_MISC_STAT	R	0h	Core 1 Misc Masked Interrupt Status - Each bit in this register corresponds to the miscellaneous interrupt (evnt_pend, stat_pend, host_pend, mdio_linkint, mdio_userint) that is enabled and generating an interrupt on C1_MISC_PULSE .

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

13.11.35 C2_RX_THRESH_STAT Register (offset = 60h) [reset = 0h]

SUBSYSTEM CORE 2 RX THRESHOLD MASKED INT STATUS REGISTER

C2_RX_THRESH_STAT is shown in Figure 13-232 and described in Table 13-245.

Figure 13-232. C2_RX_THRESH_STAT Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								C2_RX_THRESH_STAT							
																								R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 13-245. C2_RX_THRESH_STAT Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	C2_RX_THRESH_STAT	R	0h	Core 2 Receive Threshold Masked Interrupt Status - Each bit in this register corresponds to the bit in the receive threshold interrupt that is enabled and generating an interrupt on C2_RX_THRESH_PULSE.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Chapter 14

SPRUH73–July 2011

Pulse-Width Modulation Subsystem (PWMSS)

This chapter describes the PWMSS of the device.

Topic **Page**

14.1 Pulse-Width Modulation Subsystem (PWMSS) 1864
14.2 Enhanced High-Resolution PWM (eHRPWM) Module 1868
14.3 Enhanced Capture (eCAP) Module 1969
14.4 Enhanced Quadrature Encoder Pulse (eQEP) Module 2008

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.1 Pulse-Width Modulation Subsystem (PWMSS)

14.1.1 Introduction

14.1.1.1 Features

The general features of the PWMSS are:

eHRPWM

- Dedicated 16 bit time-base with Period / Frequency control
- Can support 2 independent PWM outputs with Single edge operation
- Can support 2 independent PWM outputs with Dual edge symmetric operation
- Can support 1 independent PWM output with Dual edge asymmetric operation
- Supports Dead-band generation with independent Rising and Falling edge delay control
- Provides asynchronous over-ride control of PWM signals during fault conditions
- Supports “trip zone” allocation of both latched and un-latched fault conditions
- Allows events to trigger both CPU interrupts and start of ADC conversions
- Support PWM chopping by high frequency carrier signal, used for pulse transformer gate drives.
- High-resolution module with programmable delay line.
  - Programmable on a per PWM period basis.
  - Can be inserted either on the rising edge or falling edge of the PWM pulse or both or not at all.

eCAP

- Dedicated input Capture pin
- 32 bit Time Base (counter)
- 4 x 32 bit Time-stamp Capture registers (CAP1-CAP4)
- 4 stage sequencer (Mod4 counter) which is synchronized to external events (ECAPx pin edges)
- Independent Edge polarity (Rising / Falling edge) selection for all 4 events
- Input Capture signal pre-scaling (from 1 to 16)
- One-shot compare register (2 bits) to freeze captures after 1 to 4 Time-stamp events
- Control for continuous Time-stamp captures using a 4 deep circular buffer (CAP1-CAP4) scheme
- Interrupt capabilities on any of the 4 capture events

eQEP

- Input Synchronization
- Three Stage/Six Stage Digital Noise Filter
- Quadrature Decoder Unit
- Position Counter and Control unit for position measurement
- Quadrature Edge Capture unit for low speed measurement
- Unit Time base for speed/frequency measurement
- Watchdog Timer for detecting stalls

14.1.1.2 UnsupportedFeatures

The PWMSS module features not supported are shown in Table 14-1.

Table 14-1. Unsupported Features

Feature	Reason
ePWM inputs	Not pinned out
ePWM tripzone 1-5 inputs	Only Tripzone0 is pinned out
ePWM digital comparators	Inputs not connected

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Table 14-1. Unsupported Features (continued)

Feature	Reason
eQEP quatrature input	Only output signals are connected

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.1.2 Integration

The Pulse Width Modulation Subsystem (PWMSS) includes a single instance of the Enhanced High Resolution Pulse Width Modulator (eHRPWM), Enhanced Capture (eCAP), and Enhanced Quadrature Encoded Pulse (eQEP) modules. This includes three instantiations of the PWMSS.

Figure 14-1. PWMSS Integration

14.1.2.1 PWMSS Connectivity Attributes

The general connectivity attributes for the PWMSS module are shown in Table 14-2.

Table 14-2. PWMSS Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_L4LS_GCLK
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	2 ePWM Interrupts epwm_intr_intr - Event interrupt epwm_tz_intr – Tripzone interrupt 1 eCAP interrupt ecap_intr – Capture/PWM event interrupt 1 eQEP Interrupt eqep_intr_intr – Event interrupt All interrupts go to MPU Subsystem and PRUSS, except eQEP1 and eQEP2 interrupts (MPU Subsystem only)
DMA Requests	None
Physical Address	L4 Peripheral slave port

14.1.2.2 PWMSS Clock and Reset Management

The PWMSS controllers have separate bus interface and functional clocks.

Table 14-3. PWMSS Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
PWMSS_ocp_clk Interface / Functional clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_l4ls_gclk from PRCM

14.1.2.3 PWMSS Pin list

The external signals for the PWMSS module are shown in the following table.

Table 14-4. PWMSS Pin List

Pin	Type*	Description
EPWMxA	O	PWM output A
EPWMxB	O	PWM output B
EPWM_SYNCIN	I	PWM Sync input
EPWM_SYNCOUT	O	PWM Sync output
EPWM_TRIPZONE[5:0]	I	PWM Tripzone inputs
ECAP_CAPIN_APWMOUT	I/O	eCAP Capture input / PWM output
EQEP_A	I/O	eQEP Quadrature input/output
EQEP_B	I/O	eQEP Quadrature input/output

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Table 14-4. PWMSS Pin List (continued)

Pin	Type*	Description
EQEP_INDEX	I/O	eQEP Index input/output
EQEP_STROBE	I/O	eQEP Strobe input/output

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2 Enhanced High-Resolution PWM (eHRPWM) Module

14.2.1 Introduction

An effective PWM peripheral must be able to generate complex pulse width waveforms with minimal CPU overhead or intervention. It needs to be highly programmable and very flexible while being easy to understand and use. The ePWM unit described here addresses these requirements by allocating all needed timing and control resources on a per PWM channel basis. Cross coupling or sharing of resources has been avoided; instead, the ePWM is built up from smaller single channel modules with separate resources and that can operate together as required to form a system. This modular approach results in an orthogonal architecture and provides a more transparent view of the peripheral structure, helping users to understand its operation quickly.

In this chapter, the letter x within a signal or module name is used to indicate a generic ePWM instance on a device. For example, output signals EPWMxA and EPWMxB refer to the output signals from the ePWMx instance. Thus, EPWM1A and EPWM1B belong to ePWM1 and, likewise, EPWM4A and EPWM4B belong to ePWM4.

14.2.1.1 Submodule Overview

The ePWM module represents one complete PWM channel composed of two PWM outputs: EPWMxA and EPWMxB. Multiple ePWM modules are instanced within a device as shown in Figure 14-2. Each ePWM instance is identical with one exception. Some instances include a hardware extension that allows more precise control of the PWM outputs. This extension is the high-resolution pulse width modulator (HRPWM) and is described in Section 14.2.2.10. See your device-specific data manual to determine which ePWM instances include this feature. Each ePWM module is indicated by a numerical value starting with 1. For example ePWM1 is the first instance and ePWM3 is the 3rd instance in the system and ePWMx indicates any instance.

The ePWM modules are chained together via a clock synchronization scheme that allows them to operate as a single system when required. Additionally, this synchronization scheme can be extended to the capture peripheral modules (eCAP). The number of modules is device-dependent and based on target application needs. Modules can also operate stand-alone.

Each ePWM module supports the following features:

- Dedicated 16-bit time-base counter with period and frequency control
- Two PWM outputs (EPWMxA and EPWMxB) that can be used in the following configurations::
  - Two independent PWM outputs with single-edge operation
  - Two independent PWM outputs with dual-edge symmetric operation
  - One independent PWM output with dual-edge asymmetric operation
- Asynchronous override control of PWM signals through software.
- Programmable phase-control support for lag or lead operation relative to other ePWM modules.
- Hardware-locked (synchronized) phase relationship on a cycle-by-cycle basis.
- Dead-band generation with independent rising and falling edge delay control.
- Programmable trip zone allocation of both cycle-by-cycle trip and one-shot trip on fault conditions.
- A trip condition can force either high, low, or high-impedance state logic levels at PWM outputs.
- Programmable event prescaling minimizes CPU overhead on interrupts.
- PWM chopping by high-frequency carrier signal, useful for pulse transformer gate drives.

Each ePWM module is connected to the input/output signals shown in Figure 14-2. The signals are described in detail in subsequent sections.

The order in which the ePWM modules are connected may differ from what is shown in Figure 14-2. See Section 14.2.2.3.3.2 for the synchronization scheme for a particular device. Each ePWM module consists of seven submodules and is connected within a system via the signals shown in Figure 14-3.

DRAFT COPY

Copyright © 2011, Texas Instruments Incorporated

Figure 14-2. Multiple ePWM Modules

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 14-3. Submodules and Signal Connections for an ePWM Module

Figure 14-4 shows more internal details of a single ePWM module. The main signals used by the ePWM module are:

- **PWM output signals (EPWMxA and EPWMxB).** The PWM output signals are made available external to the device through the GPIO peripheral described in the system control and interrupts guide for your device.
- **Trip-zone signals ($\overline{TZ1}$ to $\overline{TZn}$).** These input signals alert the ePWM module of an external fault condition. Each module on a device can be configured to either use or ignore any of the trip-zone signals. The trip-zone signal can be configured as an asynchronous input through the GPIO peripheral. See your device-specific data manual to determine how many trip-zone pins are available in the device.
- **Time-base synchronization input (EPWMxSYNCI) and output (EPWMxSYNCO) signals.** The synchronization signals daisy chain the ePWM modules together. Each module can be configured to either use or ignore its synchronization input. The clock synchronization input and output signal are brought out to pins only for ePWM1 (ePWM module #1). The synchronization output for ePWM1 (EPWM1SYNCO) is also connected to the SYNCI of the first enhanced capture module (eCAP1).
- **Peripheral Bus.** The peripheral bus is 32-bits wide and allows both 16-bit and 32-bit writes to the ePWM register file.

Figure 14-4 also shows the key internal submodule interconnect signals. Each submodule is described in detail in .

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 14-4. ePWM Submodules and Critical Internal Signal Interconnects

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.1.2 Register Mapping

Table 14-5 shows the complete ePWM module control and status register set grouped by submodule. Each register set is duplicated for each instance of the ePWM module. The start address for each ePWM register file instance on a device is specified in the appropriate data manual.

Table 14-5. ePWM Module Control and Status Registers Grouped by Submodule

Acronym	Offset [1]	Size (×16)	Shadow	Register Description
Time-Base Submodule Registers				
TBCTL	0h	1	No	Time-Base Control Register
TBSTS	2h	1	No	Time-Base Status Register
TBPHSHR	4h	1	No	Extension for HRPWM Phase Register [2]
TBPHS	6h	1	No	Time-Base Phase Register
TBCNT	8h	1	No	Time-Base Counter Register
TBPRD	Ah	1	Yes	Time-Base Period Register
Counter-Compare Submodule Registers				
CMPCTL	Eh	1	No	Counter-Compare Control Register
CMPAHR	10h	1	No	Extension for HRPWM Counter-Compare A Register [2]
CMPA	12h	1	Yes	Counter-Compare A Register
CMPB	14h	1	Yes	Counter-Compare B Register
Action-Qualifier Submodule Registers				
AQCTLA	16h	1	No	Action-Qualifier Control Register for Output A (EPWMxA)
AQCTLB	18h	1	No	Action-Qualifier Control Register for Output B (EPWMxB)
AQSFRC	1Ah	1	No	Action-Qualifier Software Force Register
AQCSFRC	1Ch	1	Yes	Action-Qualifier Continuous S/W Force Register Set
Dead-Band Generator Submodule Registers				
DBCTL	1Eh	1	No	Dead-Band Generator Control Register
DBRED	20h	1	No	Dead-Band Generator Rising Edge Delay Count Register
DBFED	22h	1	No	Dead-Band Generator Falling Edge Delay Count Register
PWM-Chopper Submodule Registers				
PCCTL	3Ch	1	No	PWM-Chopper Control Register
Trip-Zone Submodule Registers				
TZSEL	24h	1	No	Trip-Zone Select Register
TZCTL	28h	1	No	Trip-Zone Control Register
TZEINT	2Ah	1	No	Trip-Zone Enable Interrupt Register
TZFLG	2Ch	1	No	Trip-Zone Flag Register
TZCLR	2Eh	1	No	Trip-Zone Clear Register
TZFRC	30h	1	No	Trip-Zone Force Register
Event-Trigger Submodule Registers				
ETSEL	32h	1	No	Event-Trigger Selection Register
ETPS	34h	1	No	Event-Trigger Pre-Scale Register
ETFLG	36h	1	No	Event-Trigger Flag Register
ETCLR	38h	1	No	Event-Trigger Clear Register
ETFRC	3Ah	1	No	Event-Trigger Force Register
High-Resolution PWM (HRPWM) Submodule Registers				
HRCNFG	1040h	1	No	HRPWM Configuration Register [2]

(1) Locations not shown are reserved.

(2) These registers are only available on ePWM instances that include the high-resolution PWM (HRPWM) extension; otherwise, these locations are reserved. See your device-specific data manual to determine which instances include the HRPWM.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.2 Functional Description

Seven submodules are included in every ePWM peripheral. There are some instances that include a high-resolution submodule that allows more precise control of the PWM outputs. Each of these submodules performs specific tasks that can be configured by software.

14.2.2.1 Overview

Table 14-6 lists the eight key submodules together with a list of their main configuration parameters. For example, if you need to adjust or control the duty cycle of a PWM waveform, then you should see the counter-compare submodule in Section 14.2.2.4 for relevant details.

Table 14-6. Submodule Configuration Parameters

Submodule	Configuration Parameter or Option
Time-base (TB)	• Scale the time-base clock (TBCLK) relative to the system clock (SYSCLKOUT). • Configure the PWM time-base counter (TBCNT) frequency or period. • Set the mode for the time-base counter: – count-up mode: used for asymmetric PWM – count-down mode: used for asymmetric PWM – count-up-and-down mode: used for symmetric PWM • Configure the time-base phase relative to another ePWM module. • Synchronize the time-base counter between modules through hardware or software. • Configure the direction (up or down) of the time-base counter after a synchronization event. • Configure how the time-base counter will behave when the device is halted by an emulator. • Specify the source for the synchronization output of the ePWM module: – Synchronization input signal – Time-base counter equal to zero – Time-base counter equal to counter-compare B (CMPB) – No output synchronization signal generated.
Counter-compare (CC)	• Specify the PWM duty cycle for output EPWMxA and/or output EPWMxB • Specify the time at which switching events occur on the EPWMxA or EPWMxB output
Action-qualifier (AQ)	• Specify the type of action taken when a time-base or counter-compare submodule event occurs: – No action taken – Output EPWMxA and/or EPWMxB switched high – Output EPWMxA and/or EPWMxB switched low – Output EPWMxA and/or EPWMxB toggled • Force the PWM output state through software control • Configure and control the PWM dead-band through software
Dead-band (DB)	• Control of traditional complementary dead-band relationship between upper and lower switches • Specify the output rising-edge-delay value • Specify the output falling-edge delay value • Bypass the dead-band module entirely. In this case the PWM waveform is passed through without modification.
PWM-chopper (PC)	• Create a chopping (carrier) frequency. • Pulse width of the first pulse in the chopped pulse train. • Duty cycle of the second and subsequent pulses. • Bypass the PWM-chopper module entirely. In this case the PWM waveform is passed through without modification.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-6. Submodule Configuration Parameters (continued)

Submodule	Configuration Parameter or Option
Trip-zone (TZ)	• Configure the ePWM module to react to one, all, or none of the trip-zone pins. • Specify the tripping action taken when a fault occurs: – Force EPWMxA and/or EPWMxB high – Force EPWMxA and/or EPWMxB low – Force EPWMxA and/or EPWMxB to a high-impedance state – Configure EPWMxA and/or EPWMxB to ignore any trip condition. • Configure how often the ePWM will react to each trip-zone pin: – One-shot – Cycle-by-cycle • Enable the trip-zone to initiate an interrupt. • Bypass the trip-zone module entirely.
Event-trigger (ET)	• Enable the ePWM events that will trigger an interrupt. • Specify the rate at which events cause triggers (every occurrence or every second or third occurrence) • Poll, set, or clear event flags
High-Resolution PWM (HRPWM)	• Enable extended time resolution capabilities • Configure finer time granularity control or edge positioning

Code examples are provided in the remainder of this chapter that show how to implement various ePWM module configurations. These examples use the constant definitions shown in Example 14-1.

Example 14-1. Constant Definitions Used in the Code Examples

```
// TBCTL (Time-Base Control)
// = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
// TBCNT MODE bits
#define         TB_COUNT_UP        0x0
#define         TB_COUNT_DOWN      0x1
#define         TB_COUNT_UPDOWN    0x2
#define         TB_FREEZE          0x3
// PHSEN bit
#define         TB_DISABLE         0x0
#define         TB_ENABLE          0x1
// PRDLD bit
#define         TB_SHADOW          0x0
#define         TB_IMMEDIATE       0x1
// SYNCOSEL bits
#define         TB_SYNC_IN         0x0
#define         TB_CTR_ZERO        0x1
#define         TB_CTR_CMPB        0x2
#define         TB_SYNC_DISABLE    0x3
// HSPCLKDIV and CLKDIV bits
#define         TB_DIV1            0x0
#define         TB_DIV2            0x1
#define         TB_DIV4            0x2
// PHSDIR bit
#define         TB_DOWN            0x0
#define         TB_UP              0x1
// CMPCTL (Compare Control)
// = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
// LOADAMODE and LOADBMODE bits
#define         CC_CTR_ZERO        0x0
#define         CC_CTR_PRD         0x1
#define         CC_CTR_ZERO_PRD    0x2
#define         CC_LD_DISABLE      0x3
```


Copyright © 2011, Texas Instruments Incorporated

Example 14-1. Constant Definitions Used in the Code Examples (continued)

```
// SHDWAMODE and SHDWBMODE bits
#define         CC_SHADOW          0x0
#define         CC_IMMEDIATE       0x1
// AQCTLA and AQCTLB (Action-qualifier Control)
// = = = = = = = = = = = = = = = = = = = = = = = = = = = =
// ZRO, PRD, CAU, CAD, CBU, CBD bits
#define         AQ_NO_ACTION       0x0
#define         AQ_CLEAR           0x1
#define         AQ_SET             0x2
#define         AQ_TOGGLE          0x3
// DBCTL (Dead-Band Control)
// = = = = = = = = = = = = = = = = = = = = = = = = = = = =
// MODE bits
#define         DB_DISABLE         0x0
#define         DBA_ENABLE         0x1
#define         DBB_ENABLE         0x2
    #define DB_FULL_ENABLE 0x3
// POLSEL bits
#define         DB_ACTV_HI         0x0
#define         DB_ACTV_LOC        0x1
#define         DB_ACTV_HIC        0x2
#define         DB_ACTV_LO         0x3
// PCCTL (chopper control)
// = = = = = = = = = = = = = = = = = = = = = = = = = = = =
// CHPEN bit
#define         CHP_ENABLE         0x0
#define CHP_DISABLE 0x1
// CHPFREQ bits
#define         CHP_DIV1           0x0
#define         CHP_DIV2           0x1
#define         CHP_DIV3           0x2
#define         CHP_DIV4           0x3
#define         CHP_DIV5           0x4
#define         CHP_DIV6           0x5
#define         CHP_DIV7           0x6
#define         CHP_DIV8           0x7
// CHPDUTY bits
#define         CHP1_8TH           0x0
#define         CHP2_8TH           0x1
#define         CHP3_8TH           0x2
#define         CHP4_8TH           0x3
#define         CHP5_8TH           0x4
#define         CHP6_8TH           0x5
#define         CHP7_8TH           0x6
// TZSEL (Trip-zone Select)
// = = = = = = = = = = = = = = = = = = = = = = = = = = = =
// CBCn and OSHTn bits
#define         TZ_ENABLE          0x0
#define         TZ_DISABLE         0x1
// TZCTL (Trip-zone Control)
// = = = = = = = = = = = = = = = = = = = = = = = = = = = =
// TZA and TZB bits
#define         TZ_HIZ             0x0
#define         TZ_FORCE_HI        0x1
#define         TZ_FORCE_LO        0x2
#define         TZ_DISABLE         0x3
// ETSEL (Event-trigger Select)
// = = = = = = = = = = = = = = = = = = = = = = = = = = = =
// INTSEL, SOCASEL, SOCBSEL bits
#define         ET_CTR_ZERO        0x1
#define         ET_CTR_PRD         0x2
#define         ET_CTRU_CMPA       0x4
#define         ET_CTRD_CMPA       0x5
#define         ET_CTRU_CMPB       0x6
```

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Example 14-1. Constant Definitions Used in the Code Examples (continued)

```
#define         ET_CTRD_CMPB       0x7
// ETPS (Event-trigger Prescale)
// = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
// INTPRD, SOCAPRD, SOCBPRD bits
#define         ET_DISABLE         0x0
#define         ET_1ST             0x1
#define         ET_2ND             0x2
#define         ET_3RD             0x3
```

14.2.2.2 Proper Interrupt Initialization Procedure

When the ePWM peripheral clock is enabled it may be possible that interrupt flags may be set due to spurious events due to the ePWM registers not being properly initialized. The proper procedure for initializing the ePWM peripheral is:

1. Disable global interrupts (CPU INTM flag)
2. Disable ePWM interrupts
3. Initialize peripheral registers
4. Clear any spurious ePWM flags
5. Enable ePWM interrupts
6. Enable global interrupts

14.2.2.3 Time-Base (TB) Submodule

Each ePWM module has its own time-base submodule that determines all of the event timing for the ePWM module. Built-in synchronization logic allows the time-base of multiple ePWM modules to work together as a single system. Figure 14-5 illustrates the time-base module's place within the ePWM.

Figure 14-5. Time-Base Submodule Block Diagram

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.2.3.1 Purpose of the Time-Base Submodule

You can configure the time-base submodule for the following:

- Specify the ePWM time-base counter (TBCNT) frequency or period to control how often events occur.
- Manage time-base synchronization with other ePWM modules.
- Maintain a phase relationship with other ePWM modules.
- Set the time-base counter to count-up, count-down, or count-up-and-down mode.
- Generate the following events:
  - CTR = PRD: Time-base counter equal to the specified period (TBCNT = TBPRD) .
  - CTR = 0: Time-base counter equal to zero (TBCNT = 0000h).
- Configure the rate of the time-base clock; a prescaled version of the CPU system clock (SYSCLKOUT). This allows the time-base counter to increment/decrement at a slower rate.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.2.3.2 Controlling and Monitoring the Time-Base Submodule

Table 14-7 lists the registers used to control and monitor the time-base submodule.

Table 14-7. Time-Base Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
TBCTL	Time-Base Control Register	0h	No
TBSTS	Time-Base Status Register	2h	No
TBPHSHR	HRPWM extension Phase Register [(1)]	4h	No
TBPHS	Time-Base Phase Register	6h	No
TBCNT	Time-Base Counter Register	8h	No
TBPRD	Time-Base Period Register	Ah	Yes

(1) This register is available only on ePWM instances that include the high-resolution extension (HRPWM). On ePWM modules that do not include the HRPWM, this location is reserved. See your device-specific data manual to determine which ePWM instances include this feature.

Figure 14-6 shows the critical signals and registers of the time-base submodule. Table 14-8 provides descriptions of the key signals associated with the time-base submodule.

Figure 14-6. Time-Base Submodule Signals and Registers

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-8. Key Time-Base Signals

Signal	Description
EPWMxSYNCI	Time-base synchronization input. Input pulse used to synchronize the time-base counter with the counter of ePWM module earlier in the synchronization chain. An ePWM peripheral can be configured to use or ignore this signal. For the first ePWM module (EPWM1) this signal comes from a device pin. For subsequent ePWM modules this signal is passed from another ePWM peripheral. For example, EPWM2SYNCI is generated by the ePWM1 peripheral, EPWM3SYNCI is generated by ePWM2 and so forth. See Section 14.2.2.3.3.2 for information on the synchronization order of a particular device.
EPWMxSYNCO	Time-base synchronization output. This output pulse is used to synchronize the counter of an ePWM module later in the synchronization chain. The ePWM module generates this signal from one of three event sources: 1. EPWMxSYNCI (Synchronization input pulse) 2. CTR = 0: The time-base counter equal to zero (TBCNT = 0000h). 3. CTR = CMPB: The time-base counter equal to the counter-compare B (TBCNT = CMPB) register.
CTR = PRD	Time-base counter equal to the specified period. This signal is generated whenever the counter value is equal to the active period register value. That is when TBCNT = TBPRD.
CTR = 0	Time-base counter equal to zero. This signal is generated whenever the counter value is zero. That is when TBCNT equals 0000h.
CTR = CMPB	Time-base counter equal to active counter-compare B register (TBCNT = CMPB). This event is generated by the counter-compare submodule and used by the synchronization out logic.
CTR_dir	Time-base counter direction. Indicates the current direction of the ePWM's time-base counter. This signal is high when the counter is increasing and low when it is decreasing.
CTR_max	Time-base counter equal max value. (TBCNT = FFFFh) Generated event when the TBCNT value reaches its maximum value. This signal is only used only as a status bit.
TBCLK	Time-base clock. This is a prescaled version of the system clock (SYSCLKOUT) and is used by all submodules within the ePWM. This clock determines the rate at which time-base counter increments or decrements.

14.2.2.3.3 Calculating PWM Period and Frequency

The frequency of PWM events is controlled by the time-base period (TBPRD) register and the mode of the time-base counter. Figure 14-7 shows the period (T_{pwm}) and frequency (F_{pwm}) relationships for the up-count, down-count, and up-down-count time-base counter modes when when the period is set to 4 (TBPRD = 4). The time increment for each step is defined by the time-base clock (TBCLK) which is a prescaled version of the system clock (SYSCLKOUT).

The time-base counter has three modes of operation selected by the time-base control register (TBCTL):

- **Up-Down-Count Mode:** In up-down-count mode, the time-base counter starts from zero and increments until the period (TBPRD) value is reached. When the period value is reached, the time-base counter then decrements until it reaches zero. At this point the counter repeats the pattern and begins to increment.
- **Up-Count Mode:** In this mode, the time-base counter starts from zero and increments until it reaches the value in the period register (TBPRD). When the period value is reached, the time-base counter resets to zero and begins to increment once again.
- **Down-Count Mode:** In down-count mode, the time-base counter starts from the period (TBPRD) value and decrements until it reaches zero. When it reaches zero, the time-base counter is reset to the period value and it begins to decrement once again.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 14-7. Time-Base Frequency and Period

14.2.2.3.3.1 Time-Base Period Shadow Register

The time-base period register (TBPRD) has a shadow register. Shadowing allows the register update to be synchronized with the hardware. The following definitions are used to describe all shadow registers in the ePWM module:

- **Active Register:** The active register controls the hardware and is responsible for actions that the hardware causes or invokes.
- **Shadow Register:** The shadow register buffers or provides a temporary holding location for the active register. It has no direct effect on any control hardware. At a strategic point in time the shadow register's content is transferred to the active register. This prevents corruption or spurious operation due to the register being asynchronously modified by software.

The memory address of the shadow period register is the same as the active register. Which register is written to or read from is determined by the TBCTL[PRDLD] bit. This bit enables and disables the TBPRD shadow register as follows:

- **Time-Base Period Shadow Mode:** The TBPRD shadow register is enabled when TBCTL[PRDLD] = 0. Reads from and writes to the TBPRD memory address go to the shadow register. The shadow register contents are transferred to the active register (TBPRD (Active) ← TBPRD (shadow)) when the time-base counter equals zero (TBCNT = 0000h). By default the TBPRD shadow register is enabled.
- **Time-Base Period Immediate Load Mode:** If immediate load mode is selected (TBCTL[PRDLD] = 1), then a read from or a write to the TBPRD memory address goes directly to the active register.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.2.3.3.2 Time-Base Counter Synchronization

A time-base synchronization scheme connects all of the ePWM modules on a device. Each ePWM module has a synchronization input (EPWMxSYNCI) and a synchronization output (EPWMxSYNCO). The input synchronization for the first instance (ePWM1) comes from an external pin. The possible synchronization connections for the remaining ePWM modules is shown in Figure 14-8.

Figure 14-8. Time-Base Counter Synchronization Scheme 1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Each ePWM module can be configured to use or ignore the synchronization input. If the TBCTL[PHSEN] bit is set, then the time-base counter (TBCNT) of the ePWM module will be automatically loaded with the phase register (TBPHS) contents when one of the following conditions occur:

- **EPWMxSYNCI: Synchronization Input Pulse:** The value of the phase register is loaded into the counter register when an input synchronization pulse is detected (TBPHS → TBCNT). This operation occurs on the next valid time-base clock (TBCLK) edge.
- **Software Forced Synchronization Pulse:** Writing a 1 to the TBCTL[SWFSYNC] control bit invokes a software forced synchronization. This pulse is ORed with the synchronization input signal, and therefore has the same effect as a pulse on EPWMxSYNCI.

This feature enables the ePWM module to be automatically synchronized to the time base of another ePWM module. Lead or lag phase control can be added to the waveforms generated by different ePWM modules to synchronize them. In up-down-count mode, the TBCTL[PSHDIR] bit configures the direction of the time-base counter immediately after a synchronization event. The new direction is independent of the direction prior to the synchronization event. The TBPHS bit is ignored in count-up or count-down modes. See Figure 14-9 through Figure 14-12 for examples.

Clearing the TBCTL[PHSEN] bit configures the ePWM to ignore the synchronization input pulse. The synchronization pulse can still be allowed to flow-through to the EPWMxSYNCO and be used to synchronize other ePWM modules. In this way, you can set up a master time-base (for example, ePWM1) and downstream modules (ePWM2 - ePWMx) may elect to run in synchronization with the master.

14.2.2.3.4 Phase Locking the Time-Base Clocks of Multiple ePWM Modules

The TBCLKSYNC bit in the chip configuration register 1 (CFGCHIP1) in the System Module can be used to globally synchronize the time-base clocks of all enabled ePWM modules on a device. The TBCLKSYNC bit is part of the chip configuration registers and is described in the device-specific data manual. When TBCLKSYNC = 0, the time-base clock of all ePWM modules is stopped (default). When TBCLKSYNC = 1, all ePWM time-base clocks are started with the rising edge of TBCLK aligned. For perfectly synchronized TBCLKs, the prescaler bits in the TBCTL register of each ePWM module must be set identically. The proper procedure for enabling the ePWM clocks is as follows:

1. Enable the ePWM module clocks.
2. Set TBCLKSYNC = 0. This will stop the time-base clock within any enabled ePWM module.
3. Configure the prescaler values and desired ePWM modes.
4. Set TBCLKSYNC = 1.

14.2.2.3.5 Time-Base Counter Modes and Timing Waveforms

The time-base counter operates in one of four modes:

- Up-count mode which is asymmetrical.
- Down-count mode which is asymmetrical.
- Up-down-count which is symmetrical.
- Frozen where the time-base counter is held constant at the current value.

To illustrate the operation of the first three modes, Figure 14-9 to Figure 14-12 show when events are generated and how the time-base responds to an EPWMxSYNCI signal.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 14-9. Time-Base Up-Count Mode Waveforms

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 14-10. Time-Base Down-Count Mode Waveforms

Figure 14-11. Time-Base Up-Down-Count Waveforms, TBCTL[PHSDIR = 0] Count Down on Synchronization Event

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 14-12. Time-Base Up-Down Count Waveforms, TBCTL[PHSDIR = 1] Count Up on Synchronization Event

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.2.4 Counter-Compare (CC) Submodule

Figure 14-13 illustrates the counter-compare submodule within the ePWM. Figure 14-14 shows the basic structure of the counter-compare submodule.

Figure 14-13. Counter-Compare Submodule

Figure 14-14. Counter-Compare Submodule Signals and Registers

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.2.4.1 Purpose of the Counter-Compare Submodule

The counter-compare submodule takes as input the time-base counter value. This value is continuously compared to the counter-compare A (CMPA) and counter-compare B (CMPB) registers. When the time-base counter is equal to one of the compare registers, the counter-compare unit generates an appropriate event.

The counter-compare submodule:

- Generates events based on programmable time stamps using the CMPA and CMPB registers
  - CTR = CMPA: Time-base counter equals counter-compare A register (TBCNT = CMPA).
  - CTR = CMPB: Time-base counter equals counter-compare B register (TBCNT = CMPB)
- Controls the PWM duty cycle if the action-qualifier submodule is configured appropriately
- Shadows new compare values to prevent corruption or glitches during the active PWM cycle

14.2.2.4.2 Controlling and Monitoring the Counter-Compare Submodule

Table 14-9 lists the registers used to control and monitor the counter-compare submodule. Table 14-10 lists the key signals associated with the counter-compare submodule.

Table 14-9. Counter-Compare Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
CMPCTL	Counter-Compare Control Register.	Eh	No
CMPAHR	HRPWM Counter-Compare A Extension Register [(1)]	10h	Yes
CMPA	Counter-Compare A Register	12h	Yes
CMPB	Counter-Compare B Register	14h	Yes

(1) This register is available only on ePWM modules with the high-resolution extension (HRPWM). On ePWM modules that do not include the HRPWM, this location is reserved. Refer to the device-specific data manual to determine which ePWM instances include this feature.

Table 14-10. Counter-Compare Submodule Key Signals

Signal	Description of Event	Registers Compared
CTR = CMPA	Time-base counter equal to the active counter-compare A value	TBCNT = CMPA
CTR = CMPB	Time-base counter equal to the active counter-compare B value	TBCNT = CMPB
CTR = PRD	Time-base counter equal to the active period. Used to load active counter-compare A and B registers from the shadow register	TBCNT = TBPRD
CTR = 0	Time-base counter equal to zero. Used to load active counter-compare A and B registers from the shadow register	TBCNT = 0000h

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.2.4.3 Operational Highlights for the Counter-Compare Submodule

The counter-compare submodule is responsible for generating two independent compare events based on two compare registers:

1. CTR = CMPA: Time-base counter equal to counter-compare A register (TBCNT = CMPA).
2. CTR = CMPB: Time-base counter equal to counter-compare B register (TBCNT = CMPB).

For up-count or down-count mode, each event occurs only once per cycle. For up-down-count mode each event occurs twice per cycle, if the compare value is between 0000h and TBPRD; and occurs once per cycle, if the compare value is equal to 0000h or equal to TBPRD. These events are fed into the action-qualifier submodule where they are qualified by the counter direction and converted into actions if enabled. Refer to Section 14.2.2.5.1 for more details.

The counter-compare registers CMPA and CMPB each have an associated shadow register. Shadowing provides a way to keep updates to the registers synchronized with the hardware. When shadowing is used, updates to the active registers only occurs at strategic points. This prevents corruption or spurious operation due to the register being asynchronously modified by software. The memory address of the active register and the shadow register is identical. Which register is written to or read from is determined by the CMPCTL[SHDWAMODE] and CMPCTL[SHDWBMODE] bits. These bits enable and disable the CMPA shadow register and CMPB shadow register respectively. The behavior of the two load modes is described below:

- **Shadow Mode:** The shadow mode for the CMPA is enabled by clearing the CMPCTL[SHDWAMODE] bit and the shadow register for CMPB is enabled by clearing the CMPCTL[SHDWBMODE] bit. Shadow mode is enabled by default for both CMPA and CMPB.

  If the shadow register is enabled then the content of the shadow register is transferred to the active register on one of the following events:

  – CTR = PRD: Time-base counter equal to the period (TBCNT = TBPRD).
  – CTR = 0: Time-base counter equal to zero (TBCNT = 0000h)
  – Both CTR = PRD and CTR = 0

  Which of these three events is specified by the CMPCTL[LOADAMODE] and CMPCTL[LOADBMODE] register bits. Only the active register contents are used by the counter-compare submodule to generate events to be sent to the action-qualifier.
- **Immediate Load Mode:** If immediate load mode is selected (TBCTL[SHADWAMODE] = 1 or TBCTL[SHADWBMODE] = 1), then a read from or a write to the register will go directly to the active register.

14.2.2.4.4 Count Mode Timing Waveforms

The counter-compare module can generate compare events in all three count modes:

- Up-count mode: used to generate an asymmetrical PWM waveform.
- Down-count mode: used to generate an asymmetrical PWM waveform.
- Up-down-count mode: used to generate a symmetrical PWM waveform.

To best illustrate the operation of the first three modes, the timing diagrams in Figure 14-15 to Figure 14-18 show when events are generated and how the EPWMxSYNCI signal interacts.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 14-15. Counter-Compare Event Waveforms in Up-Count Mode

NOTE: An EPWMxSYNCI external synchronization event can cause a discontinuity in the TBCNT count sequence. This can lead to a compare event being skipped. This skipping is considered normal operation and must be taken into account.

Figure 14-16. Counter-Compare Events in Down-Count Mode

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 14-17. Counter-Compare Events in Up-Down-Count Mode, TBCTL[PHSDIR = 0] Count Down on Synchronization Event

Figure 14-18. Counter-Compare Events in Up-Down-Count Mode, TBCTL[PHSDIR = 1] Count Up on Synchronization Event

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.2.5 Action-Qualifier (AQ) Submodule

Figure 14-19 shows the action-qualifier (AQ) submodule (see shaded block) in the ePWM system. The action-qualifier submodule has the most important role in waveform construction and PWM generation. It decides which events are converted into various action types, thereby producing the required switched waveforms at the EPWMxA and EPWMxB outputs.

Figure 14-19. Action-Qualifier Submodule

14.2.2.5.1 Purpose of the Action-Qualifier Submodule

The action-qualifier submodule is responsible for the following:

- Qualifying and generating actions (set, clear, toggle) based on the following events:
  - CTR = PRD: Time-base counter equal to the period (TBCNT = TBPRD)
  - CTR = 0: Time-base counter equal to zero (TBCNT = 0000h)
  - CTR = CMPA: Time-base counter equal to the counter-compare A register (TBCNT = CMPA)
  - CTR = CMPB: Time-base counter equal to the counter-compare B register (TBCNT = CMPB)
- Managing priority when these events occur concurrently
- Providing independent control of events when the time-base counter is increasing and when it is decreasing.

14.2.2.5.2 Controlling and Monitoring the Action-Qualifier Submodule

Table 14-11 lists the registers used to control and monitor the action-qualifier submodule.

Table 14-11. Action-Qualifier Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
AQCTLA	Action-Qualifier Control Register For Output A (EPWMxA)	16h	No
AQCTLB	Action-Qualifier Control Register For Output B (EPWMxB)	18h	No
AQSFRC	Action-Qualifier Software Force Register	1Ah	No
AQCSFRC	Action-Qualifier Continuous Software Force	1Ch	Yes

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

The action-qualifier submodule is based on event-driven logic. It can be thought of as a programmable cross switch with events at the input and actions at the output, all of which are software controlled via the set of registers shown in Figure 14-20. The possible input events are summarized again in Table 14-12.

Figure 14-20. Action-Qualifier Submodule Inputs and Outputs

Table 14-12. Action-Qualifier Submodule Possible Input Events

Signal	Description	Registers Compared
CTR = PRD	Time-base counter equal to the period value	TBCNT = TBPRD
CTR = 0	Time-base counter equal to zero	TBCNT = 0000h
CTR = CMPA	Time-base counter equal to the counter-compare A	TBCNT = CMPA
CTR = CMPB	Time-base counter equal to the counter-compare B	TBCNT = CMPB
Software forced event	Asynchronous event initiated by software	

The software forced action is a useful asynchronous event. This control is handled by registers AQSFRC and AQCSFRC.

The action-qualifier submodule controls how the two outputs EPWMxA and EPWMxB behave when a particular event occurs. The event inputs to the action-qualifier submodule are further qualified by the counter direction (up or down). This allows for independent action on outputs on both the count-up and count-down phases.

The possible actions imposed on outputs EPWMxA and EPWMxB are:

- **Set High:** Set output EPWMxA or EPWMxB to a high level.
- **Clear Low:** Set output EPWMxA or EPWMxB to a low level.
- **Toggle:** If EPWMxA or EPWMxB is currently pulled high, then pull the output low. If EPWMxA or EPWMxB is currently pulled low, then pull the output high.
- **Do Nothing:** Keep outputs EPWMxA and EPWMxB at same level as currently set. Although the "Do Nothing" option prevents an event from causing an action on the EPWMxA and EPWMxB outputs, this event can still trigger interrupts. See the event-trigger submodule description in Section 14.2.2.9 for details.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Actions are specified independently for either output (EPWMxA or EPWMxB). Any or all events can be configured to generate actions on a given output. For example, both CTR = CMPA and CTR = CMPB can operate on output EPWMxA. All qualifier actions are configured via the control registers found at the end of this section.

For clarity, the drawings in this chapter use a set of symbolic actions. These symbols are summarized in Figure 14-21. Each symbol represents an action as a marker in time. Some actions are fixed in time (zero and period) while the CMPA and CMPB actions are moveable and their time positions are programmed via the counter-compare A and B registers, respectively. To turn off or disable an action, use the "Do Nothing option"; it is the default at reset.

Figure 14-21. Possible Action-Qualifier Actions for EPWMxA and EPWMxB Outputs

S/W force	TB Counter equals:				Actions
	Zero	Comp A	Comp B	Period	
SW ╳	Z ╳	CA ╳	CB ╳	P ╳	Do Nothing
SW ↓	Z ↓	CA ↓	CB ↓	P ↓	Clear Low
SW ↑	Z ↑	CA ↑	CB ↑	P ↑	Set High
SW T	Z T	CA T	CB T	P T	Toggle

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.2.5.3 Action-Qualifier Event Priority

It is possible for the ePWM action qualifier to receive more than one event at the same time. In this case events are assigned a priority by the hardware. The general rule is events occurring later in time have a higher priority and software forced events always have the highest priority. The event priority levels for up-down-count mode are shown in Table 14-13. A priority level of 1 is the highest priority and level 7 is the lowest. The priority changes slightly depending on the direction of TBCNT.

Table 14-13. Action-Qualifier Event Priority for Up-Down-Count Mode

Priority Level	Event if TBCNT is Incrementing TBCNT = 0 up to TBCNT = TBPRD	Event if TBCNT is Decrementing TBCNT = TBPRD down to TBCNT = 1
1 (Highest)	Software forced event	Software forced event
2	Counter equals CMPB on up-count (CBU)	Counter equals CMPB on down-count (CBD)
3	Counter equals CMPA on up-count (CAU)	Counter equals CMPA on down-count (CAD)
4	Counter equals zero	Counter equals period (TBPRD)
5	Counter equals CMPB on down-count (CBD) [(1)]	Counter equals CMPB on up-count (CBU) [(1)]
6 (Lowest)	Counter equals CMPA on down-count (CAD) [(1)]	Counter equals CMPA on up-count (CBU) [(1)]

(1) To maintain symmetry for up-down-count mode, both up-events (CAU/CBU) and down-events (CAD/CBD) can be generated for TBPRD. Otherwise, up-events can occur only when the counter is incrementing and down-events can occur only when the counter is decrementing.

Table 14-14 shows the action-qualifier priority for up-count mode. In this case, the counter direction is always defined as up and thus down-count events will never be taken.

Table 14-14. Action-Qualifier Event Priority for Up-Count Mode

Priority Level	Event
1 (Highest)	Software forced event
2	Counter equal to period (TBPRD)
3	Counter equal to CMPB on up-count (CBU)
4	Counter equal to CMPA on up-count (CAU)
5 (Lowest)	Counter equal to Zero

Table 14-15 shows the action-qualifier priority for down-count mode. In this case, the counter direction is always defined as down and thus up-count events will never be taken.

Table 14-15. Action-Qualifier Event Priority for Down-Count Mode

Priority Level	Event
1 (Highest)	Software forced event
2	Counter equal to Zero
3	Counter equal to CMPB on down-count (CBD)
4	Counter equal to CMPA on down-count (CAD)
5 (Lowest)	Counter equal to period (TBPRD)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

It is possible to set the compare value greater than the period. In this case the action will take place as shown in Table 14-16.

Table 14-16. Behavior if CMPA/CMPB is Greater than the Period

Counter Mode	Compare on Up-Count Event CAU/CBU	Compare on Down-Count Event CAU/CBU
Up-Count Mode	If CMPA/CMPB ≤ TBPRD period, then the event occurs on a compare match (TBCNT = CMPA or CMPB). If CMPA/CMPB > TBPRD, then the event will not occur.	Never occurs.
Down-Count Mode	Never occurs.	If CMPA/CMPB < TBPRD, the event will occur on a compare match (TBCNT = CMPA or CMPB). If CMPA/CMPB ≥ TBPRD, the event will occur on a period match (TBCNT = TBPRD).
Up-Down-Count Mode	If CMPA/CMPB < TBPRD and the counter is incrementing, the event occurs on a compare match (TBCNT = CMPA or CMPB). If CMPA/CMPB is ≥ TBPRD, the event will occur on a period match (TBCNT = TBPRD).	If CMPA/CMPB < TBPRD and the counter is decrementing, the event occurs on a compare match (TBCNT = CMPA or CMPB). If CMPA/CMPB ≥ TBPRD, the event occurs on a period match (TBCNT = TBPRD).

14.2.2.5.4 Waveforms for Common Configurations

NOTE: The waveforms in this chapter show the ePWMs behavior for a static compare register value. In a running system, the active compare registers (CMPA and CMPB) are typically updated from their respective shadow registers once every period. The user specifies when the update will take place; either when the time-base counter reaches zero or when the time-base counter reaches period. There are some cases when the action based on the new value can be delayed by one period or the action based on the old value can take effect for an extra period. Some PWM configurations avoid this situation. These include, but are not limited to, the following:

Use up-down-count mode to generate a symmetric PWM:

- If you load CMPA/CMPB on zero, then use CMPA/CMPB values greater than or equal to 1.
- If you load CMPA/CMPB on period, then use CMPA/CMPB values less than or equal to TBPRD - 1.

  This means there will always be a pulse of at least one TBCLK cycle in a PWM period which, when very short, tend to be ignored by the system.

Use up-down-count mode to generate an asymmetric PWM:

- To achieve 50%-0% asymmetric PWM use the following configuration: Load CMPA/CMPB on period and use the period action to clear the PWM and a compare-up action to set the PWM. Modulate the compare value from 0 to TBPRD to achieve 50%-0% PWM duty.

When using up-count mode to generate an asymmetric PWM:

- To achieve 0-100% asymmetric PWM use the following configuration: Load CMPA/CMPB on TBPRD. Use the Zero action to set the PWM and a compare-up action to clear the PWM. Modulate the compare value from 0 to TBPRD+1 to achieve 0-100% PWM duty.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 14-22 shows how a symmetric PWM waveform can be generated using the up-down-count mode of the TBCNT. In this mode 0%-100% DC modulation is achieved by using equal compare matches on the up count and down count portions of the waveform. In the example shown, CMPA is used to make the comparison. When the counter is incrementing the CMPA match will pull the PWM output high. Likewise, when the counter is decrementing the compare match will pull the PWM signal low. When CMPA = 0, the PWM signal is low for the entire period giving the 0% duty waveform. When CMPA = TBPRD, the PWM signal is high achieving 100% duty.

When using this configuration in practice, if you load CMPA/CMPB on zero, then use CMPA/CMPB values greater than or equal to 1. If you load CMPA/CMPB on period, then use CMPA/CMPB values less than or equal to TBPRD-1. This means there will always be a pulse of at least one TBCLK cycle in a PWM period which, when very short, tend to be ignored by the system.

Figure 14-22. Up-Down-Count Mode Symmetrical Waveform

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

The PWM waveforms in Figure 14-23 through Figure 14-28 show some common action-qualifier configurations. Some conventions used in the figures are as follows:

- TBPRD, CMPA, and CMPB refer to the value written in their respective registers. The active register, not the shadow register, is used by the hardware.
- CMPx, refers to either CMPA or CMPB.
- EPWMxA and EPWMxB refer to the output signals from ePWMx
- Up-Down means Count-up-and-down mode, Up means up-count mode and Dwn means down-count mode
- Sym = Symmetric, Asym = Asymmetric

Table 14-17 and Table 14-18 contains initialization and runtime register configurations for the waveforms in Figure 14-23.

Figure 14-23. Up, Single Edge Asymmetric Waveform, With Independent Modulation on EPWMxA and EPWMxB—Active High

(1) PWM period = (TBPRD + 1) × T_{TBCLK}

(2) Duty modulation for EPWMxA is set by CMPA, and is active high (that is, high time duty proportional to CMPA).

(3) Duty modulation for EPWMxB is set by CMPB and is active high (that is, high time duty proportional to CMPB).

(4) The "Do Nothing" actions (X) are shown for completeness, but will not be shown on subsequent diagrams.

(5) Actions at zero and period, although appearing to occur concurrently, are actually separated by one TBCLK period. TBCNT wraps from period to 0000h.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-17. EPWMx Initialization for Figure 14-23

Register	Bit	Value	Comments
TBPRD	TBPRD	600 (258h)	Period = 601 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCNT	TBCNT	0	Clear TB counter
TBCTL	CTRMODE	TB_UP	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_DISABLE	
	HSPCLKDIV	TB_DIV1	TBCLK = SYSCLK
	CLKDIV	TB_DIV1	
CMPA	CMPA	350 (15Eh)	Compare A = 350 TBCLK counts
CMPB	CMPB	200 (C8h)	Compare B = 200 TBCLK counts
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	ZRO	AQ_SET	
	CAU	AQ_CLEAR	
AQCTLB	ZRO	AQ_SET	
	CBU	AQ_CLEAR	

Table 14-18. EPWMx Run Time Changes for Figure 14-23

Register	Bit	Value	Comments
CMPA	CMPA	Duty1A	Adjust duty for output EPWM1A
CMPB	CMPB	Duty1B	Adjust duty for output EPWM1B

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-19 and Table 14-20 contains initialization and runtime register configurations for the waveforms in Figure 14-24.

Figure 14-24. Up, Single Edge Asymmetric Waveform With Independent Modulation on EPWMxA and EPWMxB—Active Low

(1) PWM period = (TBPRD + 1) × T_{TBCLK}

(2) Duty modulation for EPWMxA is set by CMPA, and is active low (that is, the low time duty is proportional to CMPA).

(3) Duty modulation for EPWMxB is set by CMPB and is active low (that is, the low time duty is proportional to CMPB).

(4) The Do Nothing actions (X) are shown for completeness here, but will not be shown on subsequent diagrams.

(5) Actions at zero and period, although appearing to occur concurrently, are actually separated by one TBCLK period. TBCNT wraps from period to 0000h.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-19. EPWMx Initialization for Figure 14-24

Register	Bit	Value	Comments
TBPRD	TBPRD	600 (258h)	Period = 601 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCNT	TBCNT	0	Clear TB counter
TBCTL	CTRMODE	TB_UP	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_DISABLE	
	HSPCLKDIV	TB_DIV1	TBCLK = SYSCLK
	CLKDIV	TB_DIV1	
CMPA	CMPA	350 (15Eh)	Compare A = 350 TBCLK counts
CMPB	CMPB	200 (C8h)	Compare B = 200 TBCLK counts
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	PRD	AQ_CLEAR	
	CAU	AQ_SET	
AQCTLB	PRD	AQ_CLEAR	
	CBU	AQ_SET	

Table 14-20. EPWMx Run Time Changes for Figure 14-24

Register	Bit	Value	Comments
CMPA	CMPA	Duty1A	Adjust duty for output EPWM1A
CMPB	CMPB	Duty1B	Adjust duty for output EPWM1B

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-21 and Table 14-22 contains initialization and runtime register configurations for the waveforms Figure 14-25. Use the code in Example 14-1 to define the headers.

Figure 14-25. Up-Count, Pulse Placement Asymmetric Waveform With Independent Modulation on EPWMxA

(1) PWM frequency = 1/((TBPRD + 1) × T_{TBCLK})

(2) Pulse can be placed anywhere within the PWM cycle (0000h - TBPRD)

(3) High time duty proportional to (CMPB - CMPA)

(4) EPWMxB can be used to generate a 50% duty square wave with frequency = 1/2 × ((TBPRD + 1) × TBCLK)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-21. EPWMx Initialization for Figure 14-25

Register	Bit	Value	Comments
TBPRD	TBPRD	600 (258h)	Period = 601 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCNT	TBCNT	0	Clear TB counter
TBCTL	CTRMODE	TB_UP	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_DISABLE	
	HSPCLKDIV	TB_DIV1	TBCLK = SYSCLK
	CLKDIV	TB_DIV1	
CMPA	CMPA	200 (C8h)	Compare A = 200 TBCLK counts
CMPB	CMPB	400 (190h)	Compare B = 400 TBCLK counts
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	CAU	AQ_SET	
	CBU	AQ_CLEAR	
AQCTLB	ZRO	AQ_TOGGLE	

Table 14-22. EPWMx Run Time Changes for Figure 14-25

Register	Bit	Value	Comments
CMPA	CMPA	EdgePosA	Adjust duty for output EPWM1A
CMPB	CMPB	EdgePosB	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-23 and Table 14-24 contains initialization and runtime register configurations for the waveforms in Figure 14-26. Use the code in Example 14-1 to define the headers.

Figure 14-26. Up-Down-Count, Dual Edge Symmetric Waveform, With Independent Modulation on EPWMxA and EPWMxB — Active Low

(1) PWM period = 2 x TBPRD × T_{TBCLK}

(2) Duty modulation for EPWMxA is set by CMPA, and is active low (that is, the low time duty is proportional to CMPA).

(3) Duty modulation for EPWMxB is set by CMPB and is active low (that is, the low time duty is proportional to CMPB).

(4) Outputs EPWMxA and EPWMxB can drive independent power switches

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-23. EPWMx Initialization for Figure 14-26

Register	Bit	Value	Comments
TBPRD	TBPRD	600 (258h)	Period = 601 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCNT	TBCNT	0	Clear TB counter
TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_DISABLE	
	HSPCLKDIV	TB_DIV1	TBCLK = SYSCLK
	CLKDIV	TB_DIV1	
CMPA	CMPA	400 (190h)	Compare A = 400 TBCLK counts
CMPB	CMPB	500 (1F4h)	Compare B = 500 TBCLK counts
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	CAU	AQ_SET	
	CAD	AQ_CLEAR	
AQCTLB	CBU	AQ_SET	
	CBD	AQ_CLEAR	

Table 14-24. EPWMx Run Time Changes for Figure 14-26

Register	Bit	Value	Comments
CMPA	CMPA	Duty1A	Adjust duty for output EPWM1A
CMPB	CMPB	Duty1B	Adjust duty for output EPWM1B

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-25 and Table 14-26 contains initialization and runtime register configurations for the waveforms in Figure 14-27. Use the code in Example 14-1 to define the headers.

Figure 14-27. Up-Down-Count, Dual Edge Symmetric Waveform, With Independent Modulation on EPWMxA and EPWMxB — Complementary

(1) PWM period = 2 × TBPRD × T_{TBCLK}

(2) Duty modulation for EPWMxA is set by CMPA, and is active low, i.e., low time duty proportional to CMPA

(3) Duty modulation for EPWMxB is set by CMPB and is active high, i.e., high time duty proportional to CMPB

(4) Outputs EPWMx can drive upper/lower (complementary) power switches

(5) Dead-band = CMPB - CMPA (fully programmable edge placement by software). Note the dead-band module is also available if the more classical edge delay method is required.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-25. EPWMx Initialization for Figure 14-27

Register	Bit	Value	Comments
TBPRD	TBPRD	600 (258h)	Period = 601 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCNT	TBCNT	0	Clear TB counter
TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_DISABLE	
	HSPCLKDIV	TB_DIV1	TBCLK = SYSCLK
	CLKDIV	TB_DIV1	
CMPA	CMPA	350 (15Eh)	Compare A = 350 TBCLK counts
CMPB	CMPB	400 (190h)	Compare B = 400 TBCLK counts
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	CAU	AQ_SET	
	CAD	AQ_CLEAR	
AQCTLB	CBU	AQ_CLEAR	
	CBD	AQ_SET	

Table 14-26. EPWMx Run Time Changes for Figure 14-27

Register	Bit	Value	Comments
CMPA	CMPA	Duty1A	Adjust duty for output EPWM1A
CMPB	CMPB	Duty1B	Adjust duty for output EPWM1B

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-27 and Table 14-28 contains initialization and runtime register configurations for the waveforms in Figure 14-28. Use the code in Example 14-1 to define the headers.

Figure 14-28. Up-Down-Count, Dual Edge Asymmetric Waveform, With Independent Modulation on EPWMxA—Active Low

(1) PWM period = 2 × TBPRD × TBCLK

(2) Rising edge and falling edge can be asymmetrically positioned within a PWM cycle. This allows for pulse placement techniques.

(3) Duty modulation for EPWMxA is set by CMPA and CMPB.

(4) Low time duty for EPWMxA is proportional to (CMPA + CMPB).

(5) To change this example to active high, CMPA and CMPB actions need to be inverted (i.e., Set ! Clear and Clear Set).

(6) Duty modulation for EPWMxB is fixed at 50% (utilizes spare action resources for EPWMxB)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-27. EPWMx Initialization for Figure 14-28

Register	Bit	Value	Comments
TBPRD	TBPRD	600 (258h)	Period = 601 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCNT	TBCNT	0	Clear TB counter
TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_DISABLE	
	HSPCLKDIV	TB_DIV1	TBCLK = SYSCLK
	CLKDIV	TB_DIV1	
CMPA	CMPA	250 (FAh)	Compare A = 250 TBCLK counts
CMPB	CMPB	450 (1C2h)	Compare B = 450 TBCLK counts
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	CAU	AQ_SET	
	CBD	AQ_CLEAR	
AQCTLB	ZRO	AQ_CLEAR	
	PRD	AQ_SET	

Table 14-28. EPWMx Run Time Changes for Figure 14-28

Register	Bit	Value	Comments
CMPA	CMPA	EdgePosA	Adjust duty for output EPWM1A
CMPB	CMPB	EdgePosB	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.2.6 Dead-Band Generator (DB) Submodule

Figure 14-29 illustrates the dead-band generator submodule within the ePWM module.

Figure 14-29. Dead-Band Generator Submodule

14.2.2.6.1 Purpose of the Dead-Band Submodule

The "Action-qualifier (AQ) Module" section discussed how it is possible to generate the required dead-band by having full control over edge placement using both the CMPA and CMPB resources of the ePWM module. However, if the more classical edge delay-based dead-band with polarity control is required, then the dead-band generator submodule should be used.

The key functions of the dead-band generator submodule are:

- Generating appropriate signal pairs (EPWMxA and EPWMxB) with dead-band relationship from a single EPWMxA input
- Programming signal pairs for:
  - Active high (AH)
  - Active low (AL)
  - Active high complementary (AHC)
  - Active low complementary (ALC)
- Adding programmable delay to rising edges (RED)
- Adding programmable delay to falling edges (FED)
- Can be totally bypassed from the signal path (note dotted lines in diagram)

14.2.2.6.2 Controlling and Monitoring the Dead-Band Submodule

The dead-band generator submodule operation is controlled and monitored via the following registers:

Table 14-29. Dead-Band Generator Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
DBCTL	Dead-Band Control Register	1Eh	No
DBRED	Dead-Band Rising Edge Delay Count Register	20h	No
DBFED	Dead-Band Falling Edge Delay Count Register	22h	No

Copyright © 2011, Texas Instruments Incorporated

14.2.2.6.3 Operational Highlights for the Dead-Band Generator Submodule

The following sections provide the operational highlights.

The dead-band submodule has two groups of independent selection options as shown in Figure 14-30.

- **Input Source Selection:** The input signals to the dead-band module are the EPWMxA and EPWMxB output signals from the action-qualifier. In this section they will be referred to as EPWMxA In and EPWMxB In. Using the DBCTL[IN_MODE) control bits, the signal source for each delay, falling-edge or rising-edge, can be selected:
  - EPWMxA In is the source for both falling-edge and rising-edge delay. This is the default mode.
  - EPWMxA In is the source for falling-edge delay, EPWMxB In is the source for rising-edge delay.
  - EPWMxA In is the source for rising edge delay, EPWMxB In is the source for falling-edge delay.
  - EPWMxB In is the source for both falling-edge and rising-edge delay.
- **Output Mode Control:** The output mode is configured by way of the DBCTL[OUT_MODE] bits. These bits determine if the falling-edge delay, rising-edge delay, neither, or both are applied to the input signals.
- **Polarity Control:** The polarity control (DBCTL[POLSEL]) allows you to specify whether the rising-edge delayed signal and/or the falling-edge delayed signal is to be inverted before being sent out of the dead-band submodule.

Figure 14-30. Configuration Options for the Dead-Band Generator Submodule

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Although all combinations are supported, not all are typical usage modes. Table 14-30 lists some classical dead-band configurations. These modes assume that the DBCTL[IN_MODE] is configured such that EPWMxA In is the source for both falling-edge and rising-edge delay. Enhanced, or non-traditional modes can be achieved by changing the input signal source. The modes shown in Table 14-30 fall into the following categories:

- **Mode 1: Bypass both falling-edge delay (FED) and rising-edge delay (RED)** Allows you to fully disable the dead-band submodule from the PWM signal path.
- **Mode 2-5: Classical Dead-Band Polarity Settings** These represent typical polarity configurations that should address all the active high/low modes required by available industry power switch gate drivers. The waveforms for these typical cases are shown in Figure 14-31. Note that to generate equivalent waveforms to Figure 14-31, configure the action-qualifier submodule to generate the signal as shown for EPWMxA.
- **Mode 6: Bypass rising-edge-delay and Mode 7: Bypass falling-edge-delay** Finally the last two entries in Table 14-30 show combinations where either the falling-edge-delay (FED) or rising-edge-delay (RED) blocks are bypassed.

Table 14-30. Classical Dead-Band Operating Modes

		DBCTL[POLSEL]		DBCTL[OUT_MODE]	
Mode	**Mode Description** [(1)]	**S3**	**S2**	**S1**	**S0**
1	EPWMxA and EPWMxB Passed Through (No Delay)	x	x	0	0
2	Active High Complementary (AHC)	1	0	1	1
3	Active Low Complementary (ALC)	0	1	1	1
4	Active High (AH)	0	0	1	1
5	Active Low (AL)	1	1	1	1
6	EPWMxA Out = EPWMxA In (No Delay) EPWMxB Out = EPWMxA In with Falling Edge Delay	0 or 1	0 or 1	0	1
7	EPWMxA Out = EPWMxA In with Rising Edge Delay EPWMxB Out = EPWMxB In with No Delay	0 or 1	0 or 1	1	0

(1) These are classical dead-band modes and assume that DBCTL[IN_MODE] = 0,0. That is, EPWMxA in is the source for both the falling-edge and rising-edge delays. Enhanced, non-traditional modes can be achieved by changing the IN_MODE configuration.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 14-31 shows waveforms for typical cases where 0% < duty < 100%.

Figure 14-31. Dead-Band Waveforms for Typical Cases (0% < Duty < 100%)

The dead-band submodule supports independent values for rising-edge (RED) and falling-edge (FED) delays. The amount of delay is programmed using the DBRED and DBFED registers. These are 10-bit registers and their value represents the number of time-base clock, TBCLK, periods a signal edge is delayed by. For example, the formula to calculate falling-edge-delay and rising-edge-delay are:

$FED = DBFED \times T_{TBCLK}$

$RED = DBRED \times T_{TBCLK}$

Where T_{TBCLK} is the period of TBCLK, the prescaled version of SYSCLKOUT.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.2.7 PWM-Chopper (PC) Submodule

Figure 14-32 illustrates the PWM-chopper (PC) submodule within the ePWM module. The PWM-chopper submodule allows a high-frequency carrier signal to modulate the PWM waveform generated by the action-qualifier and dead-band submodules. This capability is important if you need pulse transformer-based gate drivers to control the power switching elements.

Figure 14-32. PWM-Chopper Submodule

14.2.2.7.1 Purpose of the PWM-Chopper Submodule

The key functions of the PWM-chopper submodule are:

- Programmable chopping (carrier) frequency
- Programmable pulse width of first pulse
- Programmable duty cycle of second and subsequent pulses
- Can be fully bypassed if not required

14.2.2.7.2 Controlling the PWM-Chopper Submodule

The PWM-chopper submodule operation is controlled via the register in Table 14-31.

Table 14-31. PWM-Chopper Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
PCCTL	PWM-chopper Control Register	3Ch	No

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.2.7.3 Operational Highlights for the PWM-Chopper Submodule

Figure 14-33 shows the operational details of the PWM-chopper submodule. The carrier clock is derived from SYSCLKOUT. Its frequency and duty cycle are controlled via the CHPFREQ and CHPDUTY bits in the PCCTL register. The one-shot block is a feature that provides a high energy first pulse to ensure hard and fast power switch turn on, while the subsequent pulses sustain pulses, ensuring the power switch remains on. The one-shot width is programmed via the OSHTWTH bits. The PWM-chopper submodule can be fully disabled (bypassed) via the CHPEN bit.

Figure 14-33. PWM-Chopper Submodule Signals and Registers

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.2.7.4 Waveforms

Figure 14-34 shows simplified waveforms of the chopping action only; one-shot and duty-cycle control are not shown. Details of the one-shot and duty-cycle control are discussed in the following sections.

Figure 14-34. Simple PWM-Chopper Submodule Waveforms Showing Chopping Action Only

14.2.2.7.4.1 One-Shot Pulse

The width of the first pulse can be programmed to any of 16 possible pulse width values. The width or period of the first pulse is given by:

$T_{1stpulse} = T_{SYSCLKOUT} \times 8 \times OSHTWTH$

Where $T_{SYSCLKOUT}$ is the period of the system clock (SYSCLKOUT) and OSHTWTH is the four control bits (value from 1 to 16)

Figure 14-35 shows the first and subsequent sustaining pulses.

Figure 14-35. PWM-Chopper Submodule Waveforms Showing the First Pulse and Subsequent Sustaining Pulses

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.2.7.4.2 Duty Cycle Control

Pulse transformer-based gate drive designs need to comprehend the magnetic properties or characteristics of the transformer and associated circuitry. Saturation is one such consideration. To assist the gate drive designer, the duty cycles of the second and subsequent pulses have been made programmable. These sustaining pulses ensure the correct drive strength and polarity is maintained on the power switch gate during the on period, and hence a programmable duty cycle allows a design to be tuned or optimized via software control.

Figure 14-36 shows the duty cycle control that is possible by programming the CHPDUTY bits. One of seven possible duty ratios can be selected ranging from 12.5% to 87.5%.

Figure 14-36. PWM-Chopper Submodule Waveforms Showing the Pulse Width (Duty Cycle) Control of Sustaining Pulses

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.2.8 Trip-Zone (TZ) Submodule

Figure 14-37 shows how the trip-zone (TZ) submodule fits within the ePWM module. Each ePWM module is connected to every $\overline{TZ}$ signal that are sourced from the GPIO MUX. These signals indicates external fault or trip conditions, and the ePWM outputs can be programmed to respond accordingly when faults occur. See your device-specific data manual to determine the number of trip-zone pins available for the device.

Figure 14-37. Trip-Zone Submodule

14.2.2.8.1 *Purpose of the Trip-Zone Submodule*

The key functions of the trip-zone submodule are:

- Trip inputs $\overline{TZ1}$ to $\overline{TZn}$ can be flexibly mapped to any ePWM module.
- Upon a fault condition, outputs EPWMxA and EPWMxB can be forced to one of the following:
  - High
  - Low
  - High-impedance
  - No action taken
- Support for one-shot trip (OSHT) for major short circuits or over-current conditions.
- Support for cycle-by-cycle tripping (CBC) for current limiting operation.
- Each trip-zone input pin can be allocated to either one-shot or cycle-by-cycle operation.
- Interrupt generation is possible on any trip-zone pin.
- Software-forced tripping is also supported.
- The trip-zone submodule can be fully bypassed if it is not required.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.2.8.2 Controlling and Monitoring the Trip-Zone Submodule

The trip-zone submodule operation is controlled and monitored through the following registers:

Table 14-32. Trip-Zone Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
TZSEL	Trip-Zone Select Register	24h	No
TZCTL	Trip-Zone Control Register	28h	No
TZEINT	Trip-Zone Enable Interrupt Register	2Ah	No
TZFLG	Trip-Zone Flag Register	2Ch	No
TZCLR	Trip-Zone Clear Register	2Eh	No
TZFRC	Trip-Zone Force Register	30h	No

14.2.2.8.3 Operational Highlights for the Trip-Zone Submodule

The following sections describe the operational highlights and configuration options for the trip-zone submodule.

The trip-zone signals at pin $\overline{TZ1}$ to $\overline{TZn}$ is an active-low input signal. When the pin goes low, it indicates that a trip event has occurred. Each ePWM module can be individually configured to ignore or use each of the trip-zone pins. Which trip-zone pins are used by a particular ePWM module is determined by the TZSEL register for that specific ePWM module. The trip-zone signal may or may not be synchronized to the system clock (SYSCLKOUT). A minimum of 1 SYSCLKOUT low pulse on the $\overline{TZ}$ $\overline{n}$ inputs is sufficient to trigger a fault condition in the ePWM module. The asynchronous trip makes sure that if clocks are missing for any reason, the outputs can still be tripped by a valid event present on the $\overline{TZn}$ inputs.

The $\overline{TZ}$ $\overline{n}$ input can be individually configured to provide either a cycle-by-cycle or one-shot trip event for a ePWM module. The configuration is determined by the TZSEL[CBCn] and TZSEL[OSHTn] bits (where n corresponds to the trip pin) respectively.

- **Cycle-by-Cycle (CBC):** When a cycle-by-cycle trip event occurs, the action specified in the TZCTL register is carried out immediately on the EPWMxA and/or EPWMxB output. Table 14-33 lists the possible actions. In addition, the cycle-by-cycle trip event flag (TZFLG[CBC]) is set and a EPWMxTZINT interrupt is generated if it is enabled in the TZEINT register.

  The specified condition on the pins is automatically cleared when the ePWM time-base counter reaches zero (TBCNT = 0000h) if the trip event is no longer present. Therefore, in this mode, the trip event is cleared or reset every PWM cycle. The TZFLG[CBC] flag bit will remain set until it is manually cleared by writing to the TZCLR[CBC] bit. If the cycle-by-cycle trip event is still present when the TZFLG[CBC] bit is cleared, then it will again be immediately set.
- **One-Shot (OSHT):** When a one-shot trip event occurs, the action specified in the TZCTL register is carried out immediately on the EPWMxA and/or EPWMxB output. Table 14-33 lists the possible actions. In addition, the one-shot trip event flag (TZFLG[OST]) is set and a EPWMxTZINT interrupt is generated if it is enabled in the TZEINT register. The one-shot trip condition must be cleared manually by writing to the TZCLR[OST] bit.

The action taken when a trip event occurs can be configured individually for each of the ePWM output pins by way of the TZCTL[TZA] and TZCTL[TZB] register bits. One of four possible actions, shown in Table 14-33, can be taken on a trip event.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-33. Possible Actions On a Trip Event

TZCTL[TZA] and/or TZCTL[TZB]	EPWMxA and/or EPWMxB	Comment
0	High-Impedance	Tripped
1h	Force to High State	Tripped
2h	Force to Low State	Tripped
3h	No Change	Do Nothing. No change is made to the output.

Example 14-2. Trip-Zone Configurations

Scenario A:
A one-shot trip event on $\overline{TZ1}$ pulls both EPWM1A, EPWM1B low and also forces EPWM2A and EPWM2B high.

- Configure the ePWM1 registers as follows:
  - TZSEL[OSHT1] = 1: enables $\overline{TZ}$ as a one-shot event source for ePWM1
  - TZCTL[TZA] = 2: EPWM1A will be forced low on a trip event.
  - TZCTL[TZB] = 2: EPWM1B will be forced low on a trip event.
- Configure the ePWM2 registers as follows:
  - TZSEL[OSHT1] = 1: enables $\overline{TZ}$ as a one-shot event source for ePWM2
  - TZCTL[TZA] = 1: EPWM2A will be forced high on a trip event.
  - TZCTL[TZB] = 1: EPWM2B will be forced high on a trip event.

Scenario B:
A cycle-by-cycle event on $\overline{TZ5}$ pulls both EPWM1A, EPWM1B low.

A one-shot event on $\overline{TZ1}$ or $\overline{TZ6}$ puts EPWM2A into a high impedance state.

- Configure the ePWM1 registers as follows:
  - TZSEL[CBC5] = 1: enables $\overline{TZ5}$ as a one-shot event source for ePWM1
  - TZCTL[TZA] = 2: EPWM1A will be forced low on a trip event.
  - TZCTL[TZB] = 2: EPWM1B will be forced low on a trip event.
- Configure the ePWM2 registers as follows:
  - TZSEL[OSHT1] = 1: enables $\overline{TZ1}$ as a one-shot event source for ePWM2
  - TZSEL[OSHT6] = 1: enables $\overline{TZ6}$ as a one-shot event source for ePWM1
  - TZCTL[TZA] = 0: EPWM1A will be put into a high-impedance state on a trip event.
  - TZCTL[TZB] = 3: EPWM1B will ignore the trip event.

14.2.2.8.4 Generating Trip Event Interrupts

Figure 14-38 and Figure 14-39 illustrate the trip-zone submodule control and interrupt logic, respectively.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 14-38. Trip-Zone Submodule Mode Control Logic

Figure 14-39. Trip-Zone Submodule Interrupt Logic

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.2.9 Event-Trigger (ET) Submodule

Figure 14-40 shows the event-trigger (ET) submodule in the ePWM system. The event-trigger submodule manages the events generated by the time-base submodule and the counter-compare submodule to generate an interrupt to the CPU.

Figure 14-40. Event-Trigger Submodule

14.2.2.9.1 *Purpose of the Event-Trigger Submodule*

The key functions of the event-trigger submodule are:

- Receives event inputs generated by the time-base and counter-compare submodules
- Uses the time-base direction information for up/down event qualification
- Uses prescaling logic to issue interrupt requests at:
  - Every event
  - Every second event
  - Every third event
- Provides full visibility of event generation via event counters and flags

14.2.2.9.2 *Controlling and Monitoring the Event-Trigger Submodule*

The key registers used to configure the event-trigger submodule are shown in Table 14-34:

Table 14-34. Event-Trigger Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
ETSEL	Event-Trigger Selection Register	32h	No
ETPS	Event-Trigger Prescale Register	34h	No
ETFLG	Event-Trigger Flag Register	36h	No
ETCLR	Event-Trigger Clear Register	38h	No
ETFRC	Event-Trigger Force Register	3Ah	No

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.2.9.3 Operational Overview of the Event-Trigger Submodule

The following sections describe the event-trigger submodule's operational highlights.

Each ePWM module has one interrupt request line connected to the interrupt controller as shown in Figure 14-41.

Figure 14-41. Event-Trigger Submodule Inter-Connectivity to Interrupt Controller

The event-trigger submodule monitors various event conditions (the left side inputs to event-trigger submodule shown in Figure 14-42) and can be configured to prescale these events before issuing an Interrupt request. The event-trigger prescaling logic can issue Interrupt requests at:

- Every event
- Every second event
- Every third event

Figure 14-42. Event-Trigger Submodule Showing Event Inputs and Prescaled Outputs

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

- ETSEL—This selects which of the possible events will trigger an interrupt.
- ETPS—This programs the event prescaling options previously mentioned.
- ETFLG—These are flag bits indicating status of the selected and prescaled events.
- ETCLR—These bits allow you to clear the flag bits in the ETFLG register via software.
- ETFRC—These bits allow software forcing of an event. Useful for debugging or software intervention.

A more detailed look at how the various register bits interact with the Interrupt is shown in Figure 14-43.

Figure 14-43 shows the event-trigger's interrupt generation logic. The interrupt-period (ETPS[INTPRD]) bits specify the number of events required to cause an interrupt pulse to be generated. The choices available are:

- Do not generate an interrupt
- Generate an interrupt on every event
- Generate an interrupt on every second event
- Generate an interrupt on every third event

An interrupt cannot be generated on every fourth or more events.

Which event can cause an interrupt is configured by the interrupt selection (ETSEL[INTSEL]) bits. The event can be one of the following:

- Time-base counter equal to zero (TBCNT = 0000h).
- Time-base counter equal to period (TBCNT = TBPRD).
- Time-base counter equal to the compare A register (CMPA) when the timer is incrementing.
- Time-base counter equal to the compare A register (CMPA) when the timer is decrementing.
- Time-base counter equal to the compare B register (CMPB) when the timer is incrementing.
- Time-base counter equal to the compare B register (CMPB) when the timer is decrementing.

The number of events that have occurred can be read from the interrupt event counter (ETPS[INTCNT]) register bits. That is, when the specified event occurs the ETPS[INTCNT] bits are incremented until they reach the value specified by ETPS[INTPRD]. When ETPS[INTCNT] = ETPS[INTPRD] the counter stops counting and its output is set. The counter is only cleared when an interrupt is sent to the interrupt controller.

When ETPS[INTCNT] reaches ETPS[INTPRD], one of the following behaviors will occur:

- If interrupts are enabled, ETSEL[INTEN] = 1 and the interrupt flag is clear, ETFLG[INT] = 0, then an interrupt pulse is generated and the interrupt flag is set, ETFLG[INT] = 1, and the event counter is cleared ETPS[INTCNT] = 0. The counter will begin counting events again.
- If interrupts are disabled, ETSEL[INTEN] = 0, or the interrupt flag is set, ETFLG[INT] = 1, the counter stops counting events when it reaches the period value ETPS[INTCNT] = ETPS[INTPRD].
- If interrupts are enabled, but the interrupt flag is already set, then the counter will hold its output high until the ENTFLG[INT] flag is cleared. This allows for one interrupt to be pending while one is serviced.

Writing to the INTPRD bits will automatically clear the counter INTCNT = 0 and the counter output will be reset (so no interrupts are generated). Writing a 1 to the ETFRC[INT] bit will increment the event counter INTCNT. The counter will behave as described above when INTCNT = INTPRD. When INTPRD = 0, the counter is disabled and hence no events will be detected and the ETFRC[INT] bit is also ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 14-43. Event-Trigger Interrupt Generator

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.2.10 High-Resolution PWM (HRPWM) Submodule

Figure 14-44 shows the high-resolution PWM (HRPWM) submodule in the ePWM system. Some devices include the high-resolution PWM submodule, see your device-specific data manual to determine which ePWM instances include this feature.

Figure 14-44. HRPWM System Interface

Copyright © 2011, Texas Instruments Incorporated

14.2.2.10.1 Purpose of the High-Resolution PWM Submodule

The enhanced high-resolution pulse-width modulator (eHRPWM) extends the time resolution capabilities of the conventionally derived digital pulse-width modulator (PWM). HRPWM is typically used when PWM resolution falls below ~9-10 bits. The key features of HRPWM are:

- Extended time resolution capability
- Used in both duty cycle and phase-shift control methods
- Finer time granularity control or edge positioning using extensions to the Compare A and Phase registers
- Implemented using the A signal path of PWM, that is, on the EPWMxA output. EPWMxB output has conventional PWM capabilities

The ePWM peripheral is used to perform a function that is mathematically equivalent to a digital-to-analog converter (DAC). As shown in Figure 14-45, the effective resolution for conventionally generated PWM is a function of PWM frequency (or period) and system clock frequency.

Figure 14-45. Resolution Calculations for Conventionally Generated PWM

If the required PWM operating frequency does not offer sufficient resolution in PWM mode, you may want to consider HRPWM. As an example of improved performance offered by HRPWM, Table 14-35 shows resolution in bits for various PWM frequencies. Table 14-35 values assume a MEP step size of 180 ps. See your device-specific data manual for typical and maximum performance specifications for the MEP.

Table 14-35. Resolution for PWM and HRPWM

	Regular Resolution (PWM)		High Resolution (HRPWM)	
PWM Frequency (kHz)	Bits	%	Bits	%
20	12.3	0.0	18.1	0.000
50	11.0	0.0	16.8	0.001
100	10.0	0.1	15.8	0.002
150	9.4	0.2	15.2	0.003
200	9.0	0.2	14.8	0.004
250	8.6	0.3	14.4	0.005
500	7.6	0.5	13.8	0.007
1000	6.6	1.0	12.4	0.018
1500	6.1	1.5	11.9	0.027
2000	5.6	2.0	11.4	0.036

Although each application may differ, typical low-frequency PWM operation (below 250 kHz) may not require HRPWM. HRPWM capability is most useful for high-frequency PWM requirements of power conversion topologies such as:

- Single-phase buck, boost, and flyback
- Multi-phase buck, boost, and flyback
- Phase-shifted full bridge
- Direct modulation of D-Class power amplifiers

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.2.10.2 Architecture of the High-Resolution PWM Submodule

The HRPWM is based on micro edge positioner (MEP) technology. MEP logic is capable of positioning an edge very finely by sub-dividing one coarse system clock of a conventional PWM generator. The time step accuracy is on the order of 150 ps. The HRPWM also has a self-check software diagnostics mode to check if the MEP logic is running optimally, under all operating conditions.

Figure 14-46 shows the relationship between one coarse system clock and edge position in terms of MEP steps, which are controlled via an 8-bit field in the Compare A extension register (CMPAHR).

Figure 14-46. Operating Logic Using MEP

Number of coarse steps = integer(PWMduty * PWMperiod)
Number of MEP steps = fraction(PWMduty * PWMperiod) * (MEPScaleFactor)

16−bit CMPA register value	= number of coarse steps
16−bit CMPAHR register value	= (number of MEP steps) << 8 + 0x180 (rounding) (A)

A For MEP range and rounding adjustment.

To generate an HRPWM waveform, configure the TBM, CCM, and AQM registers as you would to generate a conventional PWM of a given frequency and polarity. The HRPWM works together with the TBM, CCM, and AQM registers to extend edge resolution, and should be configured accordingly. Although many programming combinations are possible, only a few are needed and practical.

14.2.2.10.3 Controlling and Monitoring the High-Resolution PWM Submodule

The MEP of the HRPWM is controlled by two extension registers, each 8-bits wide. These two HRPWM registers are concatenated with the 16-bit TBPHS and CMPA registers used to control PWM operation.

- TBPHSHR - Time-Base Phase High-Resolution Register
- CMPAHR - Counter-Compare A High-Resolution Register

Table 14-36 lists the registers used to control and monitor the high-resolution PWM submodule.

Table 14-36. HRPWM Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
TBPHSHR	Extension Register for HRPWM Phase	4h	No
CMPAHR	Extension Register for HRPWM Duty	10h	Yes
HRCNFG	HRPWM Configuration Register	1040h	No

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.2.10.4 Configuring the High-Resolution PWM Submodule

Once the ePWM has been configured to provide conventional PWM of a given frequency and polarity, the HRPWM is configured by programming the HRCNFG register located at offset address 1040h. This register provides configuration options for the following key operating modes:

- **Edge Mode:** The MEP can be programmed to provide precise position control on the rising edge (RE), falling edge (FE), or both edges (BE) at the same time. FE and RE are used for power topologies requiring duty cycle control, while BE is used for topologies requiring phase shifting, for example, phase shifted full bridge.
- **Control Mode:** The MEP is programmed to be controlled either from the CMPAHR register (duty cycle control) or the TBPHSHR register (phase control). RE or FE control mode should be used with CMPAHR register. BE control mode should be used with TBPHSHR register.
- **Shadow Mode:** This mode provides the same shadowing (double buffering) option as in regular PWM mode. This option is valid only when operating from the CMPAHR register and should be chosen to be the same as the regular load option for the CMPA register. If TBPHSHR is used, then this option has no effect.

14.2.2.10.5 Operational Highlights for the High-Resolution PWM Submodule

The MEP logic is capable of placing an edge in one of 255 (8 bits) discrete time steps, each of which has a time resolution on the order of 150 ps. The MEP works with the TBM and CCM registers to be certain that time steps are optimally applied and that edge placement accuracy is maintained over a wide range of PWM frequencies, system clock frequencies and other operating conditions. Table 14-37 shows the typical range of operating frequencies supported by the HRPWM.

Table 14-37. Relationship Between MEP Steps, PWM Frequency and Resolution

System (MHz)	MEP Steps Per SYSCLKOUT (1) (2) (3)	PWM Minimum (Hz) (4)	PWM Maximum (MHz)	Resolution at Maximum (Bits) (5)
50.0	111	763	2.50	11.1
60.0	93	916	3.00	10.9
70.0	79	1068	3.50	10.6
80.0	69	1221	4.00	10.4
90.0	62	1373	4.50	10.3
100.0	56	1526	5.00	10.1

(1) System frequency = SYSCLKOUT, that is, CPU clock. TBCLK = SYSCLKOUT
(2) Table data based on a MEP time resolution of 180 ps (this is an example value)
(3) MEP steps applied = $T_{SYSCLKOUT}$/180 ps in this example.
(4) PWM minimum frequency is based on a maximum period value, TBPRD = 65 535. PWM mode is asymmetrical up-count.
(5) Resolution in bits is given for the maximum PWM frequency stated.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.2.10.5.1 Edge Positioning

In a typical power control loop (switch modes, digital motor control (DMC), uninterruptible power supply (UPS)), a digital controller (PID, 2pole/2zero, lag/lead, etc.) issues a duty command, usually expressed in a per unit or percentage terms.

In the following example, assume that for a particular operating point, the demanded duty cycle is 0.405 or 40.5% on-time and the required converter PWM frequency is 1.25 MHz. In conventional PWM generation with a system clock of 100 MHz, the duty cycle choices are in the vicinity of 40.5%. In Figure 14-47, a compare value of 32 counts (duty = 40%) is the closest to 40.5% that you can attain. This is equivalent to an edge position of 320 ns instead of the desired 324 ns. This data is shown in Table 14-38.

By utilizing the MEP, you can achieve an edge position much closer to the desired point of 324 ns. Table 14-38 shows that in addition to the CMPA value, 22 steps of the MEP (CMPAHR register) will position the edge at 323.96 ns, resulting in almost zero error. In this example, it is assumed that the MEP has a step resolution of 180 ns.

Figure 14-47. Required PWM Waveform for a Requested Duty = 40.5%

Table 14-38. CMPA vs Duty (left), and [CMPA:CMPAHR] vs Duty (right)

CMPA (count) (1) (2) (3)	DUTY (%)	High Time (ns)	CMPA (count)	CMPAHR (count)	Duty (%)	High Time (ns)
28	35.0	280	32	18	40.405	323.24
29	36.3	290	32	19	40.428	323.42
30	37.5	300	32	20	40.450	323.60
31	38.8	310	32	21	40.473	323.78
32	40.0	320	32	22	40.495	323.96
33	41.3	330	32	23	40.518	324.14
34	42.5	340	32	24	40.540	324.32
			32	25	40.563	324.50
Required			32	26	40.585	324.68
32.40	40.5	324	32	27	40.608	324.86

(1) System clock, SYSCLKOUT and TBCLK = 100 MHz, 10 ns
(2) For a PWM Period register value of 80 counts, PWM Period = 80 × 10 ns = 800 ns, PWM frequency = 1/800 ns = 1.25 MHz
(3) Assumed MEP step size for the above example = 180 ps

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.2.10.5.2 Scaling Considerations

The mechanics of how to position an edge precisely in time has been demonstrated using the resources of the standard (CMPA) and MEP (CMPAHR) registers. In a practical application, however, it is necessary to seamlessly provide the CPU a mapping function from a per-unit (fractional) duty cycle to a final integer (non-fractional) representation that is written to the [CMPA:CMPAHR] register combination.

To do this, first examine the scaling or mapping steps involved. It is common in control software to express duty cycle in a per-unit or percentage basis. This has the advantage of performing all needed math calculations without concern for the final absolute duty cycle, expressed in clock counts or high time in ns. Furthermore, it makes the code more transportable across multiple converter types running different PWM frequencies.

To implement the mapping scheme, a two-step scaling procedure is required.

Assumptions for this example:

System clock, SYSCLKOUT	=	10 ns (100 MHz)
PWM frequency	=	1.25 MHz (1/800 ns)
Required PWM duty cycle, **PWMDuty**	=	0.405 (40.5%)
PWM period in terms of coarse steps, **PWMperiod** (800 ns/10 ns)	=	80
Number of MEP steps per coarse step at 180 ps (10 ns/180 ps), **MEP_SF**	=	55
Value to keep CMPAHR within the range of 1-255 and fractional rounding constant (default value)	=	180h

Step 1: Percentage Integer Duty value conversion for CMPA register

CMPA register value	=	int(**PWMDuty** × **PWMperiod**); int means integer part
	=	int(0.405 × 80)
	=	int(32.4)
CMPA register value	=	32 (20h)

Step 2: Fractional value conversion for CMPAHR register

CMPAHR register value	=	(frac(**PWMDuty** × **PWMperiod**) × **MEP_SF**) << 8) + 180h; frac means fractional part
	=	(frac(32.4) × 55 <<8) + 180h; Shift is to move the value as CMPAHR high byte
	=	((0.4 × 55) <<8) + 180h
	=	(22 <<8) + 180h
	=	22 × 256 + 180h ; Shifting left by 8 is the same multiplying by 256.
	=	5632 + 180h
	=	1600h + 180h
CMPAHR value	=	1780h; CMPAHR value = 1700h, lower 8 bits will be ignored by hardware.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.2.10.5.3 Duty Cycle Range Limitation

In high resolution mode, the MEP is not active for 100% of the PWM period. It becomes operational 3 SYSCLK cycles after the period starts.

Duty cycle range limitations are illustrated in Figure 14-48. This limitation imposes a lower duty cycle limit on the MEP. For example, precision edge control is not available all the way down to 0% duty cycle. Although for the first 3 or 6 cycles, the HRPWM capabilities are not available, regular PWM duty control is still fully operational down to 0% duty. In most applications this should not be an issue as the controller regulation point is usually not designed to be close to 0% duty cycle.

Figure 14-48. Low % Duty Cycle Range Limitation Example When PWM Frequency = 1 MHz

If the application demands HRPWM operation in the low percent duty cycle region, then the HRPWM can be configured to operate in count-down mode with the rising edge position (REP) controlled by the MEP. This is illustrated in Figure 14-49. In this case low percent duty limitation is no longer an issue.

Figure 14-49. High % Duty Cycle Range Limitation Example when PWM Frequency = 1 MHz

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.2.10.6 Scale Factor Optimizing Software (SFO)

The micro edge positioner (MEP) logic is capable of placing an edge in one of 255 discrete time steps. As previously mentioned, the size of these steps is on the order of 150 ps. The MEP step size varies based on worst-case process parameters, operating temperature, and voltage. MEP step size increases with decreasing voltage and increasing temperature and decreases with increasing voltage and decreasing temperature. Applications that use the HRPWM feature should use the TI-supplied MEP scale factor optimizer (SFO) software functions. SFO functions help to dynamically determine the number of MEP steps per SYSCLKOUT period while the HRPWM is in operation.

To utilize the MEP capabilities effectively during the Q15 duty to [CMPA:CMPAHR] mapping function (see Section 14.2.2.10.5.2), the correct value for the MEP scaling factor (MEP_SF) needs to be known by the software. To accomplish this, each HRPWM module has built in self-check and diagnostics capabilities that can be used to determine the optimum MEP_SF value for any operating condition. TI provides a C-callable library containing two SFO functions that utilize this hardware and determine the optimum MEP_SF. As such, MEP Control and Diagnostics registers are reserved for TI use.

Currently, there are two released versions of the SFO library - SFO_TI_Build.lib and SFO_TI_Build_V5.lib. Versions 2, 3, and 4 were TI-Internal only. A high-level comparison table between the two library versions can be found in Section 14.2.2.10.6.1. A detailed description of the SFO_TI_Build.lib software functions follows in Section 14.2.2.10.6.2. Information on the SFO_TI_Build_V5.lib software functions, which support up to 16 HRPWM channels, can be found in Section 14.2.2.10.6.4.

NOTE: For floating-point devices, when compiling application code for floating point (fpu32 mode), libraries utilized by the application code must also be compiled for floating point. The SFO_TI_Build_fpu.lib and SFO_TI_Build_V5_fpu.lib are available as the floating-point compiled equivalents to the fixed-point SFO_TI_Build.lib and SFO_TI_Build_V5.lib libraries. The SFO functions in the fpu-version libraries are C-code-compatible to their fixed-point equivalents.

14.2.2.10.6.1 SFO Library Version Comparison

Table 14-39 includes a high-level comparison between SFO_TI_Build.lib and SFO_TI_Build_V5.lib. A detailed description of SFO_TI_Build_V5.lib follows the table, and more information on SFO_TI_Build.lib can be found in Section 14.2.2.10.6.

Table 14-39. SFO Library Version Comparison

	SYSCLK Frequency	ePWM Frequency	SFO_TI_Build.lib	SFO_TI_Build_V5.lib	Unit
Maximum HRPWM channels supported	-	-	Up to 4	Up to 16	channels
Total static variable memory size	-	-	220	79 (1 channel) to 192 (16 channel)	words
MepEn runs on multiple channels concurrently?	-	-	Yes	No	-
Error-checking?	-	-	No	Yes	-
Typical time required for MepEn to update MEP_ScaleFactor on 1 channel if called repetitively without interrupts	-	400 kHz	3.26	1.5	seconds
	-	1 MHz	1.308	0.6	seconds
	-	2 MHz	0.66	0.3	seconds
	-	3.33 MHz	0.396	0.18	seconds
	-	200 MHz	0.066	0.03	seconds
Typical time required for MepDis to update MEP_ScaleFactor on 1 channel if called repetitively without interrupts	100 MHz	-	0.83	1.86	milliseconds
	60 MHz	-	1.38	1.38	milliseconds
	50 MHz	-	1.66	1.66	milliseconds

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.2.10.6.2 SFO_TI_Build Library Routines

Table 14-40 provides functional descriptions of the two SFO library routines in SFO_TI_Build.lib.

Table 14-40. SFO Library Routines

Function	Description
SFO_MepDis(n)	**Scale Factor Optimizer with MEP Disabled** This routine runs faster, as the calibration logic works when HRPWM capabilities are disabled; therefore, HRPWM capabilities cannot be run concurrently when the ePWMn is being used. The function returns a value in the variable array: MEP_ScaleFactor[n][1] = Number of MEP steps/SYSCLKOUT If TBCLK is not equal to SYSCLKOUT, then the returned value must be adjusted to reflect the correct TBCLK: MEP steps per TBCLK = MEP_ScaleFactor[n] × (SYSCLKOUT/TBCLK)[1] **Constraints when using this function:** MEP diagnostics logic uses SYSCLKOUT not TBCLK and hence SYSCLKOUT restriction is an important constraint. SFO_MepDis(n) function does not require a starting Scale Factor value. **When to use:** If one of the ePWM modules is not used in HRPWM mode, then it can be dedicated to run the SFO diagnostics for the modules that are running HRPWM mode. Here the single MEP_SF value obtained can be applied to other ePWM modules. This assumes that all HRPWM module's MEP steps are similar but may not be identical. The ePWM module that is not active in HRPWM mode is still fully operational in conventional PWM mode and can be used to drive PWM pins. The SFO function only makes use of the MEP diagnostics logic. The other ePWM modules operating in HRPWM mode incur only a 3-cycle minimum duty limitation.
SFO_MepEn(n)	**Scale Factor Optimizer with MEP Enabled** This routine runs slower as the calibration logic is used concurrently while HRPWM capabilities are being used by the ePWM module. The function returns a value in the variable array: MEP_ScaleFactor[n][1] = Number of MEP steps/SYSCLKOUT = Number of MEP steps/TBCLK **Constraints when using this function:** MEP diagnostics logic uses SYSCLKOUT not TBCLK and hence SYSCLKOUT restriction is an important constraint. SFO_MepEn(n) function does require a starting Scale Factor value.MEP_ScaleFactor[0] needs to be initialized to a typical MEP step size value. **NOTE:** SFO_MepEn(n) only supports the following HRPWM configuration: • HRCNFG[HRLOAD] = 0 (load on CTR = 0) • HRCNFG[EDGMODE] = 10 (falling edge MEP control) **When to use:** If the application requires all ePWM modules to have HRPWM capability (MEP is operational), then the SFO_MepEn(n) function should run for each of the active ePWM modules with HRPWM capability. • In the above case, a 6-cycle MEP inactivity zone exists at the start of the PWM period. See Section 14.2.2.10.5.3 on duty cycle range limitation. • If all ePWM modules are using the same TBCLK prescaler, then it is also possible to run the SFO_MepEn(n) function for only one ePWM module and to use the SFO return value for the other modules. In this case only one ePWM module incurs the 6-cycle limitation, and remaining modules incur only a 3-cycle minimum duty limitation. See "Duty cycle limitation" section. This assumes that all HRPWM module's MEP steps are similar but may not be identical.

(1) n is the ePWM module number on which the SFO function operates.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Both routines can be run as background tasks in a slow loop requiring negligible CPU cycles. In most applications only one of these routines will be needed. However, if the application has free HRPWM resources then both the routines could be used. The repetition rate at which an SFO function needs to be executed depends on the applications operating environment. As with all digital CMOS devices temperature and supply voltage variations have an effect on MEP operation. However, in most applications these parameters vary slowly and therefore it is often sufficient to execute the SFO function once every 5 to 10 seconds or so. If more rapid variations are expected, then execution may have to be performed more frequently to match the application. Note, there is no high limit restriction on the SFO function repetition rate, hence it can execute as quickly as the background loop is capable.

While using HRPWM feature with no SFO diagnostics, HRPWM logic will not be active for the first 3 TBCLK cycles of the PWM period. While running the application in this configuration, if CMPA register value is less than 3 cycles, then its CMPAHR register must be cleared to zero. This would avoid any unexpected transitions on PWM signal.

However, if SFO diagnostic function SFO_MepEn is used in the background, then HRPWM logic will not be active for the first 6 TBCLK cycles of PWM period. While using SFO_MepEn function if CMPA register value is less than 6 cycles, then its CMPAHR register must be cleared to zero. This would avoid any unexpected transitions on PWM signal. Also note that the SFO_MepDis function cannot be used concurrently with PWM signals with HRPWM enabled.

14.2.2.10.6.3 SFO_TI_Build Software Usage

Software library functions SFO_MepEn(int n) and SFO_MepDis(int n) calculate the MEP scale factor for up to four ePWM modules. The scale factor is an integer value in the range 1–255, and represents the number of micro step edge positions available for a system clock period. The scale factor value is returned in a global array of integer values called MEP_ScaleFactor[x], where x is the maximum number of HRPWM channels for a device plus one. For example, see Table 14-41.

Table 14-41. Factor Values

Software function calls	Functional description	Updated Variable MEP_ScaleFactor[5] [(1)]
SFO_MepDis(n)		
SFO_MepDis(1);	Returns the scale factor value to array index 1	MEP_ScaleFactor[1]
SFO_MepDis(2);	Returns the scale factor value to array index 2	MEP_ScaleFactor[2]
SFO_MepDis(3);	Returns the scale factor value to array index 3	MEP_ScaleFactor[3]
SFO_MepDis(4);	Returns the scale factor value to array index 4	MEP_ScaleFactor[4]
SFO_MepEn(n)		
SFO_MepEn(1);	Returns the scale factor value to array index 1	MEP_ScaleFactor[1]
SFO_MepEn(2);	Returns the scale factor value to array index 2	MEP_ScaleFactor[2]
SFO_MepEn(3);	Returns the scale factor value to array index 3	MEP_ScaleFactor[3]
SFO_MepEn(4);	Returns the scale factor value to array index 4	MEP_ScaleFactor[4]

(1) MEP_ScaleFactor[0] variable is a starting value and used by the SFO software functions internally

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

To use the HRPWM feature of the ePWMs it is recommended that the SFO functions be used as follows. For different devices that may have fewer HRPWM channels, modifications will be required in Step 1 and Step 2.

NOTE: The following example assumes there are four ePWM instances that contain the HRPWM submodule in the device. See your device-specific data manual to determine the number of ePWM instances that contain the HRPWM submodule.

Step 1. Add Include Files

The SFO.h file needs to be included as follows. This include file is mandatory while using the SFO library function. These include files are optional if customized header files are used in the end applications.

Example 14-3. A Sample of How to Add Include Files

```
#include "SFO.h"                  // SFO lib functions (needed for HRPWM)
```

Step 2. Element Declaration

Declare a 5-element array of integer variables as follows:

Example 14-4. Declaring an Element

```
int    MEP_ScaleFactor[5] = {0,0,0,0,0};  // Scale factor values for ePWM1-4
int    MEP_SF1, MEP_SF2, MEP_SF3, MEP_SF4
volatile struct EPWM_REGS *ePWM[] = {&EPwm1Regs, &EPwm1Regs, &EPwm2Regs, &EPwm3Regs, &EPwm4Regs};
```

Step 3. MEP_ScaleFactor Initialization

After power up, the SFO_MepEn(n) function needs a starting Scale Factor value. This value can be conveniently determined by using one of the ePWM modules to run the SFO_MepDis(n) function prior to configuring its PWM outputs for the application. SFO_MepDis(n) function does not require a starting Scale Factor value.

As part of the one-time initialization code, include the following:

Example 14-5. Initializing With a Scale Factor Value

```
//     MEP_ScaleFactor variables initialized using function SFO_MepDis
while (MEP_ScaleFactor[1] == 0) SFO_MepDis(1);    //SFO for HRPWM1
while (MEP_ScaleFactor[2] == 0) SFO_MepDis(2);    //SFO for HRPWM2
while (MEP_ScaleFactor[3] == 0) SFO_MepDis(3);    //SFO for HRPWM3
while (MEP_ScaleFactor[4] == 0) SFO_MepDis(4);    //SFO for HRPWM4

// Initialize a common seed variable MEP_ScaleFactor[0] // required for all SFO functions
MEP_ScaleFactor[0] = MEP_ScaleFactor[1]; //Common variable for SFOMepEN(n) function
```

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Step 4. Application Code

While the application is running, fluctuations in both device temperature and supply voltage may be expected. To be sure that optimal Scale Factors are used for each ePWM module, the SFO function should be re-run periodically as part of a slower back-ground loop. Some examples of this are shown here.

Example 14-6. SFO Function Calls

```
main()
{

    //  User code
    // Case1: ePWM1,2,3,4 are running in HRPWM mode
        SFO_MepEn(1);                       // Each of these of function enables
        SFO_MepEn(2);                       // the respective MEP diagnostic logic
        SFO_MepEn(3);                       // and returns MEP Scale factor value
        SFO_MepEn(4);


        MEP_SF1 = MEP_ScaleFactor[1];       // used for ePWM1
        MEP_SF2 = MEP_ScaleFactor[2];       // used for ePWM2
        MEP_SF3 = MEP_ScaleFactor[3];       // used for ePWM3
        MEP_SF4 = MEP_ScaleFactor[4];       // used for ePWM4
    // Case2:ePWM1,2,3 only are running in HRPWM mode.
                One of the ePWM channel(as an example ePWM4) is used as for
                Scale factor calibration
   // Here minimum duty cycle limitation is 3 clock cycles.
   //
   // HRPWM 4 MEP diagnostics circuit is used to estimate the MEP steps
   // with the assumption that all HRPWM channels behave similarly
   // though may not be identical.
        SFO_MepDis(4);                     // MEP steps using ePWM4
        MEP_SF1 = MEP_ScaleFactor[4];      // used for ePWM1
        MEP_SF2 = MEP_SF1                  // used for ePWM2
        MEP_SF3 = MEP_SF1                  // used for ePWM3
        MEP_SF4 = MEP_SF1                  // used for ePWM4
 }
```

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.2.10.6.4 SFO_TI_Build_V5 Library Routines

In SFO_TI_Build_V5.lib, the diagnostic software has been optimized to use less memory, to minimize the calibration time, and to support up to 16 HRPWM channels. Table 14-42 provides functional description of the two SFO library routines in SFO_TI_Build_V5.lib.

Table 14-42. SFO V5 Library Routines

Function	Description
int SFO_MepDis_V5 (n)	**Scale Factor Optimizer V5 with MEP Disabled** This routine is very similar to the SFO_MepDis() routine in the original SFO library, but with one change. It now returns a 1 when MEP disabled calibration is complete, or a 0 while calibration is still running. This function runs faster than the SFO_MepEn _V5() routine and cannot be used on an ePWM channel while HRPWM capabilities are enabled for that channel. If there is a spare ePWM channel available in the system, SFO_MepDis_V5() can be run for that channel, and the resulting MEP_ScaleFactor[n] value can be copied into the MEP_ScaleFactor[n] for all other channels. Because the MEP step behavior in a particular piece of silicon is similar on all HRPWM channels with regard to temperature and voltage, using one dedicated HRPWM channel for calibration by calling the SFO_MepDis_V5 function will reduce software overhead. The function returns a value in the variable array: MEP_ScaleFactor[n] = Number of MEP steps/SYSCLKOUT If TBCLK is not equal to SYSCLKOUT, then the returned value must be adjusted to reflect the correct TBCLK: MEP steps per TBCLK = MEP_ScaleFactor[n] × (SYSCLKOUT/TBCLK) **Constraints when using this function:** • MEP diagnostics logic uses SYSCLKOUT and not TBCLK. Hence, the SYSCLKOUT restriction is an important constraint. • If TBCLK does not equal SYSCLKOUT, the TBCLK frequency must be great enough so that MEP steps per TBCLK do not exceed 255. This is due to the restriction that there can be no more than 255 MEP steps in a coarse step. • This function cannot be run on an ePWM channel with HRPWM capabilities enabled. **Usage:** • If one of the ePWM modules is running in normal ePWM mode, then it can be used to run the SFO diagnostics function. Here, the single MEP_ScaleFactor value obtained for that channel can be copied and used as the MEP_ScaleFactor for the other ePWM modules which are running HRPWM mode. This assumes that all HRPWM modules' MEP steps are similar but may not be identical. • This routine returns a 1 when calibration is finished on the specified channel or a 0 if calibration is still running. • The ePWM module that is not active in HRPWM mode is still fully operational in conventional PWM mode and can be used to drive PWM pins. The SFO function only makes use of the MEP diagnostics logic in the HRPWM circuitry. • SFO_MepDis_V5(n) function does not require a starting Scale Factor value. • The other ePWM modules operating in HRPWM mode incur only a 3-cycle minimum duty cycle limitation.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-42. SFO V5 Library Routines (continued)

Function	Description
int SFO_MepEn_V5 (n)	**Scale Factor Optimizer V5 with MEP Enabled** This function runs slower than the SFO_MepDis_V5() routine and runs SFO diagnostics on an ePWM channel with HRPWM capabilities enabled for that channel. The function returns a value in the variable array: MEP_ScaleFactor[n] = Number of MEP steps/SYSCLKOUT = Number of MEP steps/TBCLK **Constraints when using this function:** • This routine must be run on one channel at a time and cannot be run on multiple channels concurrently. When it has finished updating the MEP_ScaleFactor for a channel, it will return a 1. If it is still calibrating, it will return a 0. A background loop should exist in the user code which calls SFO_MepEn_V5(n) repetitively until it returns a 1. Then the function can be called for the next channel .[(1)] **NOTE:** Unlike the original SFO_MepEn(n) routine, this routine cannot run on multiple channels concurrently. Do not call SFO_MepEn_V5(n) for another channel until the function returns a 1 for the current channel. Otherwise, the MEP_ScaleFactor for both channels will become corrupted. SFO_MepEn_V5(n) in *SFO_TI_Build_V5.lib* supports only the following HRPWM configuration: – HRCNFG[HRLOAD] = 0 (load on CTR = 0) – HRCNFG[EDGMODE] = 10 (falling edge MEP control) An upgraded version of SFO_MepEn_V5(n) in *SFO_TI_Build_V5B.lib* supports all available HRPWM configurations. When using this version, the HRCNFG register must be initialized with the appropriate configuration after calling SFO_MepDis_V5(n) to seed the MEP_ScaleFactor[n] and prior to calling SFO_MepEn_V5(n). • MEP diagnostics logic uses SYSCLKOUT and not TBCLK. Hence, the SYSCLKOUT restriction is an important constraint. **Usage:** • After calling SFO_MepDis(n) to seed MEP_ScaleFactor[n], and prior to using the SFO_MepEn_V5(n) function in *SFO_TI_Build_V5B.lib*, the HRCNFG register must be initialized with the desired HRPWM configuration. Otherwise, calibration will not be initiated, and calls to SFO_MepEn_V5(n) will continuously return zero. • The SFO_MepEn_V5(n) function requires a starting scale factor value, MEP_ScaleFactor[0]. MEP_ScaleFactor[0] needs to be initialized to a typical MEP step size value. To do this, SFO_MepDis_V5(n) can be run on an ePWM channel while the HRPWM is disabled, and the resulting MEP_ScaleFactor[n] value can be copied into MEP_ScaleFactor[0]. • If there are drastic environmental changes to your system (i.e. temperature/voltage), it is generally a good idea to re-seed MEP_ScaleFactor[0] with a new typical MEP step size value for the changed conditions. • Because SFO_MepEn_V5(n) can be run on only one channel at a time, it is only recommended for systems where there are no spare HRPWM channels available, so SFO calibration must be performed on all channels with HRPWM capabilities enabled. In this case, a 6-cycle MEP inactivity zone exists at the start of each PWM period on all HRPWM channels. See Section 14.2.2.10.5.3 on duty cycle range limitation. • The function returns: – A 1 when it has finished SFO calibration for the current channel – A 0 when SFO diagnostics are still running for the channel

(1) If SFO calibration must be run on multiple channels at a time while HRPWM capabilities are enabled, the previous version of the SFO library, SFO_TI_Build.lib, which uses more memory resources, can be used instead, and SFO_MepEn(n) can run concurrently for up to 4 ePWM channels with HRPWM enabled.

Copyright © 2011, Texas Instruments Incorporated

Table 14-42. SFO V5 Library Routines (continued)

Function	Description
	– A 2 as an error indicator after calibration has completed if the resulting MEP_ScaleFactor for the channel differs from the original MEP_ScaleFactor[0] seed value by more than +/- 15. The function must be called repetitively before it will return a 1. This takes a longer time to complete than the SFO_MepDis_V5(n) calibration. If it returns a 2, the MEP_ScaleFactor for the channel has finished updating and is outside the typical drift range of MEP_ScaleFactor[0] +/-15 even with large temperature and voltage variations. If the reason for the large difference between the seed and the channel scale factor is known and acceptable, the user may choose to ignore the return of 2, and treat it as a return value of 1, indicating that calibration is complete. Otherwise, if the large difference is unexpected, there are steps to take to remedy the error: 1. Check your code to ensure SFO_MepEn_V5(n) is not being called on more than one channel at a time. 2. If the above is not effective, run SFO_MepDis_V5(n) again and re-seed Mep_ScaleFactor[0]. 3. If neither of the above 2 steps work, there may be a system problem. The application firmware should perform a shutdown or an appropriate recovery procedure. • If all ePWM modules are using the same TBCLK prescaler, then it is possible to run the SFO_MepEn_V5(n) function for only one ePWM module and to use the MEP_ScaleFactor value for that module for the other modules also. In this case only one ePWM module incurs the 6-cycle duty limitation, and the remaining modules incur only a 3-cycle minimum duty limitation. This assumes that all HRPWM modules' MEP steps are similar but may not be identical.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.2.10.6.5 SFO_TI_Build_V5 Software Usage

Software library functions int SFO_MepEn_V5(int n) and int SFO_MepDis_V5(int n) calculate the MEP scale factor for ePWMn modules, where n= the ePWM channel number. The scale factor value, which represents the number of micro-steps available in a system clock period, is returned in a global array of integer values called MEP_ScaleFactor[x], where x is the maximum number of HRPWM channels for a device plus one. For example, if the maximum number of HRPWM channels for a device is 16, the scale factor array would be MEP_ScaleFactor[17]. Both SFO_MepEn_V5 and SFO_MepDis_V5 themselves also return a 1 when calibration has completed, indicating the MEP_ScaleFactor has been successfully updated for the channel, and a 0 when calibration is still on-going. A return of 2 represents an out-of-range error.

Table 14-43. Software Functions

Software functional calls	Functional Description
int SFO_MepDis_V5(int n)[(1)]	
status = SFO_MepDis_V5(1)	The scale factor in MEP_ScaleFactor[1] is updated when status = 1
status = SFO_MepDis_V5(2)	The scale factor in MEP_ScaleFactor[2] is updated when status = 1
...	...
status = SFP_MepDis_V5(16)	The scale factor in MEP_ScaleFactor[16] is updated when status = 1 or 2
int SFO_MepEn_V5(int n)[(1)]	
status = SFO_MepEn_V5(1);	The scale factor in MEP_ScaleFactor[1] is updated when status = 1 or 2
status = SFO_MepEn_V5(2);	The scale factor in MEP_ScaleFactor[2] is updated when status = 1 or 2
...	...
status = SFO_MepEn_V5(16);	The scale factor in MEP_ScaleFactor[16] is updated when status = 1 or 2

(1) MEP_ScaleFactor[0] is a starting seed value used by the SFO software functions internally.

To use the HRPWM feature of the ePWMs, it is recommended that the SFO functions in TI_Build_V5.lib be used as described here. For different devices that may have fewer HRPWM channels, modifications will be required in Step 1 and Step 2.

NOTE: The following example assumes there are four ePWM instances that contain the HRPWM submodule in the device. See your device-specific data manual to determine the number of ePWM instances that contain the HRPWM submodule.

Step 1. Add Include Files

The SFO_V5.h file needs to be included as follows. This include file is mandatory when using the SFO V5 library functions.

Example 14-7. A Sample of How to Add Include Files

```
#include SFO_V5.h                    // SFO V5 lib functions (needed for HRPWM)
```

Step 2. Define Number of HRPWM Channels Used

In the SFO_V5.h file, the maximum number of HRPWM's used for a particular device must be defined. PWM_CH must equal the number of HRPWM channels plus 1.

To save static variable memory, fewer than the maximum number of HRPWM channels may be defined with some caution. To do this, PWM_CH can be set to the largest ePWM channel number plus 1. For instance, if only ePWM1A and ePWM2A channels are required as HRPWM channels, PWM_CH can be set to 3. However, if only ePWM15A and ePWM16A channels are required as HRPWM channels, PWM_CH must still be set to 17.

Copyright © 2011, Texas Instruments Incorporated

Example 14-8. Defining Number of HRPWM Channels Used (Plus 1)

```
 // SFO_V5.H
// NOTE: THIS IS A VERY IMPORTANT STEP. PWM_CH MUST BE DEFINED FIRST BEFORE
//       BUILDING CODE.

#define PWM_CH 17 // Maximum of 16 HRPWM channels (16+1=17)
```

Step 3. Element Declaration

Declare an array of integer variables with a length equal to PWM_CH, and an array of pointers to EPWM register structures. The array of pointers will include pointers for up to 16 EPWM register structures plus one dummy pointer in location EPWM[0] for a device with 16 EPWM channels. Likewise, it will include pointers for up to 4 EPWM register structures plus 1 for a device with 4 EPWM registers and up to 3 EPWM register structures plus 1 for a device with 3 EPWM registers.

Example 14-9. Declaring Elements Required by SFO_TI_Build_V5.lib

```
int MEP_ScaleFactor[PWM_CH] = {0,0,0,0,0, // Scale factor values for ePWM 1-16
                               0,0,0,0, // and MEP_ScaleFactor[0]
                               0,0,0,0, // For fewer HRPWM channels, there
                               0,0,0,0};// will be fewer zeros initialized.

// Declare a volatile array of pointers to EPWM Register structures.
// Only point to registers that exist. If a device has only 6 EPWMs (PWM_CH is 7),
// the array will include pointers for up to 6 EPWM register structures plus one
// dummy pointer in the ePWM[0] location.
volatile struct EPWM_REGS *ePWM[PWM_CH]  {&EPwm1Regs, &EPwm1Regs, &EPwm2Regs,
&EPwm3Regs, &EPwm4Regs, &EPwm5Regs, &EPwm6Regs, &EPwm7Regs,
&EPwm8Regs, &EPwm9Regs, &EPwm10Regs, &EPwm11Regs, &EPwm12Regs,
&EPwm13Regs, &EPwm14Regs, &EPwm15Regs, &EPwm16Regs};
```

Step 4. MEP_ScaleFactor Initialization

After power up, the SFO_MepEn_V5(n) function needs a typical scale factor starting seed value in MEP_ScaleFactor[0]. This value can be conveniently determined using one of the ePWM modules to run the SFO_MepDis_V5(n) function prior to initializing the PWM settings for the application. The SFO_MepDis_V5(n) function does not require a starting scale factor value.

As part of the one-time initialization code, include the following:

Example 14-10. Initialization With a Scale Factor Value

```
// MEP_ScaleFactor variables initialized using function SFO_MepDis_V5
Uint16 I;
for (I=1; i<PWM_CH; I++)        // For channels 1-16
{
    while (SFO_MepDis_V5(I)==0); // Calls MepDis until MEP_ScaleFactor updated
}

// Initialize MEP_ScaleFactor[0] with a typical MEP seed value
// required for SFO_MepEn_V5
    MEP_ScaleFactor[0] = MEP_ScaleFactor[1];
```

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Step 5. Application Code

While the application is running, fluctuations in both device temperature and supply voltage may be expected. To be sure that optimal scale factors are used for each ePWM module, the SFO function should be re-run periodically as part of a slower background loop. Some examples of this are shown here.

Example 14-11. SFO Function Calls

```
main()
{
   Uint16 current_ch = 1; // keeps track of current HRPWM channel being calibrated
   Uint16 status;

  //  User code

  //  Case 1: All ePWMs are running in HRPWM mode
  //          Here, the minimum duty cycle limitation is 6 clock cycles.


        status = SFO_MepEn_V5(current_ch); // MepEn called here
        if (status ==1)                    // if MEP_ScaleFactor has been updated
        {
            current_ch++;                  // move on to the next channel
        }

        else if (status==2)                // if MEP_ScaleFactor differs from
        {                                  // MEP_ScaleFactor[0] seed by more than
             error();                      // +/-15, flag an error
        }
        if (current_ch==PWM_CH)            // if last channel has been reached
        {
            current_ch = 1;                // go back to channel 1
        }


  //  Case 2: all ePWMs except one are running in HRPWM mode.
  //          One of the ePWM channels (ePWM16 in this example) is used
  //          for SFO_MepDis_V5 scale factor calibration.
  //          Here, the minimum duty cycle limitation is 3 clock cycles.

  //          HRPWM16 diagnostics circuitry is used to estimate the MEP steps
  //          with the assumption that all HRPWM channels behave similarly
  //          though they may not be identical.

    while (SFO_MepDis_V5(16)==0); // wait until MEP_ScaleFactor[16] updated
    for (I=1; I<(PWM_CH-1); I++)        // Update scale factors for ePWM 1-15.
    {
         MEP_ScaleFactor[i] = MEP_ScaleFactor[16];
```

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.3 Use Cases

An ePWM module has all the local resources necessary to operate completely as a standalone module or to operate in synchronization with other identical ePWM modules.

14.2.3.1 Overview of Multiple Modules

Previously in this user's guide, all discussions have described the operation of a single module. To facilitate the understanding of multiple modules working together in a system, the ePWM module described in reference is represented by the more simplified block diagram shown in Figure 14-50. This simplified ePWM block shows only the key resources needed to explain how a multiswitch power topology is controlled with multiple ePWM modules working together.

Figure 14-50. Simplified ePWM Module

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.3.2 Key Configuration Capabilities

The key configuration choices available to each module are as follows:

- Options for SyncIn
  - Load own counter with phase register on an incoming sync strobe—enable (EN) switch closed
  - Do nothing or ignore incoming sync strobe—enable switch open
  - Sync flow-through - SyncOut connected to SyncIn
  - Master mode, provides a sync at PWM boundaries—SyncOut connected to CTR = PRD
  - Master mode, provides a sync at any programmable point in time—SyncOut connected to CTR = CMPB
  - Module is in standalone mode and provides No sync to other modules—SyncOut connected to X (disabled)
- Options for SyncOut
  - Sync flow-through - SyncOut connected to SyncIn
  - Master mode, provides a sync at PWM boundaries—SyncOut connected to CTR = PRD
  - Master mode, provides a sync at any programmable point in time—SyncOut connected to CTR = CMPB
  - Module is in standalone mode and provides No sync to other modules—SyncOut connected to X (disabled)

For each choice of SyncOut, a module may also choose to load its own counter with a new phase value on a SyncIn strobe input or choose to ignore it, i.e., via the enable switch. Although various combinations are possible, the two most common—master module and slave module modes—are shown in Figure 14-51.

Figure 14-51. EPWM1 Configured as a Typical Master, EPWM2 Configured as a Slave

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.3.3 Controlling Multiple Buck Converters With Independent Frequencies

One of the simplest power converter topologies is the buck. A single ePWM module configured as a master can control two buck stages with the same PWM frequency. If independent frequency control is required for each buck converter, then one ePWM module must be allocated for each converter stage. Figure 14-52 shows four buck stages, each running at independent frequencies. In this case, all four ePWM modules are configured as Masters and no synchronization is used. Figure 14-53 shows the waveforms generated by the setup shown in Figure 14-52; note that only three waveforms are shown, although there are four stages.

Figure 14-52. Control of Four Buck Stages. Here $F_{PWM1} \neq F_{PWM2} \neq F_{PWM3} \neq F_{PWM4}$

NOTE: Θ = X indicates value in phase register is a "don't care"

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 14-53. Buck Waveforms for Figure 14-52 (Note: Only three bucks shown here)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-44. EPWM1 Initialization for Figure 14-53

Register	Bit	Value	Comments
TBPRD	TBPRD	1200 (4B0h)	Period = 1201 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCTL	CTRMODE	TB_UP	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_DISABLE	
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	PRD	AQ_CLEAR	
	CAU	AQ_SET	

Table 14-45. EPWM2 Initialization for Figure 14-53

Register	Bit	Value	Comments
TBPRD	TBPRD	1400 (578h)	Period = 1401 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCTL	CTRMODE	TB_UP	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_DISABLE	
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	PRD	AQ_CLEAR	
	CAU	AQ_SET	

Table 14-46. EPWM3 Initialization for Figure 14-53

Register	Bit	Value	Comments
TBPRD	TBPRD	800 (320h)	Period = 801 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCTL	CTRMODE	TB_UP	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_DISABLE	
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	PRD	AQ_CLEAR	
	CAU	AQ_SET	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Example 14-12. Configuration for Example in Figure 14-53

```
// Run Time (Note: Example execution of one run-time instance)
//=========================================================
    EPwm1Regs.CMPA.half.CMPA = 700;              // adjust duty for output EPWM1A
    EPwm2Regs.CMPA.half.CMPA = 700;              // adjust duty for output EPWM2A
    EPwm3Regs.CMPA.half.CMPA = 500;              // adjust duty for output EPWM3A
```

14.2.3.4 Controlling Multiple Buck Converters With Same Frequencies

If synchronization is a requirement, ePWM module 2 can be configured as a slave and can operate at integer multiple (N) frequencies of module 1. The sync signal from master to slave ensures these modules remain locked. Figure 14-54 shows such a configuration; Figure 14-55 shows the waveforms generated by the configuration.

Figure 14-54. Control of Four Buck Stages. (Note: $F_{PWM2} = N \times F_{PWM1}$)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 14-55. Buck Waveforms for Figure 14-54 (Note: $F_{PWM2} = F_{PWM1}$)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-47. EPWM1 Initialization for Figure 14-54

Register	Bit	Value	Comments
TBPRD	TBPRD	600 (258h)	Period = 1200 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_CTR_ZERO	Sync down-stream module
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	CAU	AQ_SET	Set actions for EPWM1A
	CAD	AQ_CLEAR	
AQCTLB	CBU	AQ_SET	Set actions for EPWM1B
	CBD	AQ_CLEAR	

Table 14-48. EPWM2 Initialization for Figure 14-54

Register	Bit	Value	Comments
TBPRD	TBPRD	600 (258h)	Period = 1200 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_ENABLE	Phase loading enabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_IN	Sync flow-through
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	CAU	AQ_SET	Set actions for EPWM2A
	CAD	AQ_CLEAR	
AQCTLB	CBU	AQ_SET	Set actions for EPWM2B
	CBD	AQ_CLEAR	

Example 14-13. Code Snippet for Configuration in Figure 14-54

```
// Run Time (Note: Example execution of one run-time instance)
//==========================================================
     EPwm1Regs.CMPA.half.CMPA = 400;       // adjust duty for output EPWM1A
     EPwm1Regs.CMPB = 200;                 // adjust duty for output EPWM1B
     EPwm2Regs.CMPA.half.CMPA = 500;       // adjust duty for output EPWM2A
     EPwm2Regs.CMPB = 300;                 // adjust duty for output EPWM2B
```

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.3.5 Controlling Multiple Half H-Bridge (HHB) Converters

Topologies that require control of multiple switching elements can also be addressed with these same ePWM modules. It is possible to control a Half-H bridge stage with a single ePWM module. This control can be extended to multiple stages. Figure 14-56 shows control of two synchronized Half-H bridge stages where stage 2 can operate at integer multiple (N) frequencies of stage 1. Figure 14-57 shows the waveforms generated by the configuration shown in Figure 14-56.

Module 2 (slave) is configured for Sync flow-through; if required, this configuration allows for a third Half-H bridge to be controlled by PWM module 3 and also, most importantly, to remain in synchronization with master module 1.

Figure 14-56. Control of Two Half-H Bridge Stages ($F_{PWM2} = N \times F_{PWM1}$)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 14-57. Half-H Bridge Waveforms for Figure 14-56 (Note: Here $F_{PWM2} = F_{PWM1}$)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-49. EPWM1 Initialization for Figure 14-56

Register	Bit	Value	Comments
TBPRD	TBPRD	600 (258h)	Period = 1200 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_CTR_ZERO	Sync down-stream module
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	ZRO	AQ_SET	Set actions for EPWM1A
	CAU	AQ_CLEAR	
AQCTLB	ZRO	AQ_CLEAR	Set actions for EPWM1B
	CAD	AQ_SET	

Table 14-50. EPWM2 Initialization for Figure 14-56

Register	Bit	Value	Comments
TBPRD	TBPRD	600 (258h)	Period = 1200 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_ENABLE	Phase loading enabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_IN	Sync flow-through
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	ZRO	AQ_SET	Set actions for EPWM2A
	CAU	AQ_CLEAR	
AQCTLB	ZRO	AQ_CLEAR	Set actions for EPWM2B
	CAD	AQ_SET	

Example 14-14. Code Snippet for Configuration in Figure 14-56

```
// Run Time (Note: Example execution of one run-time instance)
//==========================================================
     EPwm1Regs.CMPA.half.CMPA = 400; // adjust duty for output EPWM1A
     EPwm1Regs.CMPB = 200;           // adjust duty for output EPWM1B
     EPwm2Regs.CMPA.half.CMPA = 500; // adjust duty for output EPWM2A
     EPwm2Regs.CMPB = 250;           // adjust duty for output EPWM2B
```

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.3.6 Controlling Dual 3-Phase Inverters for Motors (ACI and PMSM)

The idea of multiple modules controlling a single power stage can be extended to the 3-phase Inverter case. In such a case, six switching elements can be controlled using three PWM modules, one for each leg of the inverter. Each leg must switch at the same frequency and all legs must be synchronized. A master + two slaves configuration can easily address this requirement. Figure 14-58 shows how six PWM modules can control two independent 3-phase Inverters; each running a motor.

As in the cases shown in the previous sections, we have a choice of running each inverter at a different frequency (module 1 and module 4 are masters as in Figure 14-58), or both inverters can be synchronized by using one master (module 1) and five slaves. In this case, the frequency of modules 4, 5, and 6 (all equal) can be integer multiples of the frequency for modules 1, 2, 3 (also all equal).

Figure 14-58. Control of Dual 3-Phase Inverter Stages as Is Commonly Used in Motor Control

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 14-59. 3-Phase Inverter Waveforms for Figure 14-58 (Only One Inverter Shown)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-51. EPWM1 Initialization for Figure 14-58

Register	Bit	Value	Comments
TBPRD	TBPRD	800 (320h)	Period = 1600 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_CTR_ZERO	Sync down-stream module
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	CAU	AQ_SET	Set actions for EPWM1A
	CAD	AQ_CLEAR	
DBCTL	MODE	DB_FULL_ENABLE	Enable Dead-band module
	POLSEL	DB_ACTV_HIC	Active Hi complementary
DBFED	DBFED	50	FED = 50 TBCLKs
	DBRED	50	RED = 50 TBCLKs

Table 14-52. EPWM2 Initialization for Figure 14-58

Register	Bit	Value	Comments
TBPRD	TBPRD	800 (320h)	Period = 1600 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_ENABLE	Slave module
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_IN	Sync flow-through
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	CAU	AQ_SET	Set actions for EPWM2A
	CAD	AQ_CLEAR	
DBCTL	MODE	DB_FULL_ENABLE	Enable Dead-band module
	POLSEL	DB_ACTV_HIC	Active Hi complementary
DBFED	DBFED	50	FED = 50 TBCLKs
	DBRED	50	RED = 50 TBCLKs

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-53. EPWM3 Initialization for Figure 14-58

Register	Bit	Value	Comments
TBPRD	TBPRD	800 (320h)	Period = 1600 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_ENABLE	Slave module
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_IN	Sync flow-through
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	CAU	AQ_SET	Set actions for EPWM3A
	CAD	AQ_CLEAR	
DBCTL	MODE	DB_FULL_ENABLE	Enable Dead-band module
	POLSEL	DB_ACTV_HIC	Active Hi complementary
DBFED	DBFED	50	FED = 50 TBCLKs
	DBRED	50	RED = 50 TBCLKs

Example 14-15. Code Snippet for Configuration in Figure 14-58

```
// Run Time (Note: Example execution of one run-time instance)
//=========================================================
EPwm1Regs.CMPA.half.CMPA = 500; // adjust duty for output EPWM1A
EPwm2Regs.CMPA.half.CMPA = 600; // adjust duty for output EPWM2A
EPwm3Regs.CMPA.half.CMPA = 700; // adjust duty for output EPWM3A
```

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.3.7 Practical Applications Using Phase Control Between PWM Modules

So far, none of the examples have made use of the phase register (TBPHS). It has either been set to zero or its value has been a don't care. However, by programming appropriate values into TBPHS, multiple PWM modules can address another class of power topologies that rely on phase relationship between legs (or stages) for correct operation. As described in the TB module section, a PWM module can be configured to allow a SyncIn pulse to cause the TBPHS register to be loaded into the TBCNT register. To illustrate this concept, Figure 14-60 shows a master and slave module with a phase relationship of 120°, that is, the slave leads the master.

Figure 14-60. Configuring Two PWM Modules for Phase Control

Figure 14-61 shows the associated timing waveforms for this configuration. Here, TBPRD = 600 for both master and slave. For the slave, TBPHS = 200 (200/600 × 360° = 120°). Whenever the master generates a SyncIn pulse (CTR = PRD), the value of TBPHS = 200 is loaded into the slave TBCNT register so the slave time-base is always leading the master's time-base by 120°.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 14-61. Timing Waveforms Associated With Phase Control Between 2 Modules

14.2.3.8 Controlling a 3-Phase Interleaved DC/DC Converter

A popular power topology that makes use of phase-offset between modules is shown in Figure 14-62. This system uses three PWM modules, with module 1 configured as the master. To work, the phase relationship between adjacent modules must be F = 120°. This is achieved by setting the slave TBPHS registers 2 and 3 with values of 1/3 and 2/3 of the period value, respectively. For example, if the period register is loaded with a value of 600 counts, then TBPHS (slave 2) = 200 and TBPHS (slave 3) = 400. Both slave modules are synchronized to the master 1 module.

This concept can be extended to four or more phases, by setting the TBPHS values appropriately. The following formula gives the TBPHS values for N phases:

TBPHS(N,M) = (TBPRD/N) × (M - 1)

Where:

N = number of phases

M = PWM module number

For example, for the 3-phase case (N = 3), TBPRD = 600,

TBPHS(3,2) = (600/3) × (2 - 1) = 200 × 1 = 200 (Phase value for Slave module 2)

TBPHS(3,3) = (600/3) × (3 - 1) = 200 × 2 = 400 (Phase value for Slave module 3)

Figure 14-63 shows the waveforms for the configuration in Figure 14-62.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 14-62. Control of a 3-Phase Interleaved DC/DC Converter

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 14-63. 3-Phase Interleaved DC/DC Converter Waveforms for Figure 14-62

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-54. EPWM1 Initialization for Figure 14-62

Register	Bit	Value	Comments
TBPRD	TBPRD	450 (1C2h)	Period = 900 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_CTR_ZERO	Sync down-stream module
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	CAU	AQ_SET	Set actions for EPWM1A
	CAD	AQ_CLEAR	
DBCTL	MODE	DB_FULL_ENABLE	Enable Dead-band module
	POLSEL	DB_ACTV_HIC	Active Hi complementary
DBFED	DBFED	20	FED = 20 TBCLKs
	DBRED	20	RED = 20 TBCLKs

Table 14-55. EPWM2 Initialization for Figure 14-62

Register	Bit	Value	Comments
TBPRD	TBPRD	450 (1C2h)	Period = 900 TBCLK counts
TBPHS	TBPHS	300	Phase = (300/900) × 360 = 120°
TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_ENABLE	Slave module
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_IN	Sync flow-through
	PHSDIR	TB_DOWN	Count DOWN on sync
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	CAU	AQ_SET	Set actions for EPWM2A
	CAD	AQ_CLEAR	
DBCTL	MODE	DB_FULL_ENABLE	Enable Dead-band module
	POLSEL	DB_ACTV_HIC	Active Hi complementary
DBFED	DBFED	20	FED = 20 TBCLKs
	DBRED	20	RED = 20 TBCLKs

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-56. EPWM3 Initialization for Figure 14-62

Register	Bit	Value	Comments
TBPRD	TBPRD	450 (1C2h)	Period = 900 TBCLK counts
TBPHS	TBPHS	300	Phase = (300/900) × 360 = 120°
TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_ENABLE	Slave module
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_IN	Sync flow-through
	PHSDIR	TB_UP	Count UP on sync
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	CAU	AQ_SET	Set actions for EPWM3A
	CAD	AQ_CLEAR	
DBCTL	MODE	DB_FULL_ENABLE	Enable Dead-band module
	POLSEL	DB_ACTV_HIC	Active Hi complementary
DBFED	DBFED	20	FED = 20 TBCLKs
	DBRED	20	RED = 20 TBCLKs

Example 14-16. Code Snippet for Configuration in Figure 14-62

```
// Run Time (Note: Example execution of one run-time instance)
//==========================================================
EPwm1Regs.CMPA.half.CMPA = 285;         // adjust duty for output EPWM1A
EPwm2Regs.CMPA.half.CMPA = 285;         // adjust duty for output EPWM2A
EPwm3Regs.CMPA.half.CMPA = 285;         // adjust duty for output EPWM3A
```

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.2.3.9 Controlling Zero Voltage Switched Full Bridge (ZVSFB) Converter

The example given in Figure 14-64 assumes a static or constant phase relationship between legs (modules). In such a case, control is achieved by modulating the duty cycle. It is also possible to dynamically change the phase value on a cycle-by-cycle basis. This feature lends itself to controlling a class of power topologies known as *phase-shifted full bridge*, or *zero voltage switched full bridge.* Here the controlled parameter is not duty cycle (this is kept constant at approximately 50 percent); instead it is the phase relationship between legs. Such a system can be implemented by allocating the resources of two PWM modules to control a single power stage, which in turn requires control of four switching elements. Figure 14-65 shows a master/slave module combination synchronized together to control a full H-bridge. In this case, both master and slave modules are required to switch at the same PWM frequency. The phase is controlled by using the slave's phase register (TBPHS). The master's phase register is not used and therefore can be initialized to zero.

Figure 14-64. Controlling a Full-H Bridge Stage ($F_{PWM2} = F_{PWM1}$)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 14-65. ZVS Full-H Bridge Waveforms

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-57. EPWM1 Initialization for Figure 14-64

Register	Bit	Value	Comments
TBPRD	TBPRD	1200 (4B0h)	Period = 1201 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCTL	CTRMODE	TB_UP	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_CTR_ZERO	Sync down-stream module
CMPA	CMPA	600 (258h)	Set 50% duty for EPWM1A
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	ZRO	AQ_SET	Set actions for EPWM1A
	CAU	AQ_CLEAR	
DBCTL	MODE	DB_FULL_ENABLE	Enable Dead-band module
	POLSEL	DB_ACTV_HIC	Active Hi complementary
DBFED	DBFED	50	FED = 50 TBCLKs
	DBRED	70	RED = 70 TBCLKs

Table 14-58. EPWM2 Initialization for Figure 14-64

Register	Bit	Value	Comments
TBPRD	TBPRD	1200 (4B0h)	Period = 1201 TBCLK counts
TBPHS	TBPHS	0	Clear Phase Register to 0
TBCTL	CTRMODE	TB_UP	
	PHSEN	TB_ENABLE	Slave module
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_IN	Sync flow-through
CMPA	CMPA	600 (258h)	Set 50% duty for EPWM2A
CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on CTR = 0
	LOADBMODE	CC_CTR_ZERO	Load on CTR = 0
AQCTLA	ZRO	AQ_SET	Set actions for EPWM2A
	CAU	AQ_CLEAR	
DBCTL	MODE	DB_FULL_ENABLE	Enable Dead-band module
	POLSEL	DB_ACTV_HIC	Active Hi complementary
DBFED	DBFED	30	FED = 30 TBCLKs
	DBRED	40	RED = 40 TBCLKs

Example 14-17. Code Snippet for Configuration in Figure 14-64

```
// Run Time (Note: Example execution of one run-time instance)
//==========================================================
EPwm2Regs.TBPHS = 1200-300;                        // Set Phase reg to 300/1200 * 360 = 90 deg
EPwm1Regs.DBFED = FED1_NewValue;                   // Update ZVS transition interval
EPwm1Regs.DBRED = RED1_NewValue;                   // Update ZVS transition interval
EPwm2Regs.DBFED = FED2_NewValue;                   // Update ZVS transition interval
EPwm2Regs.DBRED = RED2_NewValue;                   // Update ZVS transition interval
```


Copyright © 2011, Texas Instruments Incorporated

14.2.4 Registers

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.3 Enhanced Capture (eCAP) Module

14.3.1 Introduction

14.3.1.1 Purpose of the Peripheral

Uses for eCAP include:

- Sample rate measurements of audio inputs
- Speed measurements of rotating machinery (for example, toothed sprockets sensed via Hall sensors)
- Elapsed time measurements between position sensor pulses
- Period and duty cycle measurements of pulse train signals
- Decoding current or voltage amplitude derived from duty cycle encoded current/voltage sensors

14.3.1.2 Features

The eCAP module includes the following features:

- 32-bit time base counter
- 4-event time-stamp registers (each 32 bits)
- Edge polarity selection for up to four sequenced time-stamp capture events
- Interrupt on either of the four events
- Single shot capture of up to four event time-stamps
- Continuous mode capture of time-stamps in a four-deep circular buffer
- Absolute time-stamp capture
- Difference (Delta) mode time-stamp capture
- All above resources dedicated to a single input pin
- When not used in capture mode, the ECAP module can be configured as a single channel PWM output

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.3.2 Functional Description

The eCAP module represents one complete capture channel that can be instantiated multiple times depending on the target device. In the context of this guide, one eCAP channel has the following independent key resources:

- Dedicated input capture pin
- 32-bit time base counter
- 4 × 32-bit time-stamp capture registers (CAP1-CAP4)
- 4-stage sequencer (Modulo4 counter) that is synchronized to external events, ECAP pin rising/falling edges.
- Independent edge polarity (rising/falling edge) selection for all 4 events
- Input capture signal prescaling (from 2-62)
- One-shot compare register (2 bits) to freeze captures after 1 to 4 time-stamp events
- Control for continuous time-stamp captures using a 4-deep circular buffer (CAP1-CAP4) scheme
- Interrupt capabilities on any of the 4 capture events

Multiple identical eCAP modules can be contained in a system as shown in Figure 14-66. The number of modules is device-dependent and is based on target application needs. In this chapter, the letter x within a signal or module name is used to indicate a generic eCAP instance on a device.

Figure 14-66. Multiple eCAP Modules

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.3.2.1 Capture and APWM Operating Mode

You can use the eCAP module resources to implement a single-channel PWM generator (with 32 bit capabilities) when it is not being used for input captures. The counter operates in count-up mode, providing a time-base for asymmetrical pulse width modulation (PWM) waveforms. The CAP1 and CAP2 registers become the active period and compare registers, respectively, while CAP3 and CAP4 registers become the period and capture shadow registers, respectively. Figure 14-67 is a high-level view of both the capture and auxiliary pulse-width modulator (APWM) modes of operation.

Figure 14-67. Capture and APWM Modes of Operation

(1) A single pin is shared between CAP and APWM functions. In capture mode, it is an input; in APWM mode, it is an output.

(2) In APWM mode, writing any value to CAP1/CAP2 active registers also writes the same value to the corresponding shadow registers CAP3/CAP4. This emulates immediate mode. Writing to the shadow registers CAP3/CAP4 invokes the shadow mode.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.3.2.2 Capture Mode Description

Figure 14-68 shows the various components that implement the capture function.

Figure 14-68. Capture Function Diagram

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.3.2.2.1 Event Prescaler

An input capture signal (pulse train) can be prescaled by N = 2-62 (in multiples of 2) or can bypass the prescaler. This is useful when very high frequency signals are used as inputs. Figure 14-69 shows a functional diagram and Figure 14-70 shows the operation of the prescale function.

Figure 14-69. Event Prescale Control

(1) When a prescale value of 1 is chosen (ECCTL1[13:9] = 0000) the input capture signal by-passes the prescale logic completely.

Figure 14-70. Prescale Function Waveforms

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.3.2.2.2 Edge Polarity Select and Qualifier

- Four independent edge polarity (rising edge/falling edge) selection multiplexers are used, one for each capture event.
- Each edge (up to 4) is event qualified by the Modulo4 sequencer.
- The edge event is gated to its respective CAP*n* register by the Mod4 counter. The CAP*n* register is loaded on the falling edge.

14.3.2.2.3 Continuous/One-Shot Control

- The Mod4 (2 bit) counter is incremented via edge qualified events (CEVT1-CEVT4).
- The Mod4 counter continues counting (0->1->2->3->0) and wraps around unless stopped.
- A 2-bit stop register is used to compare the Mod4 counter output, and when equal stops the Mod4 counter and inhibits further loads of the CAP1-CAP4 registers. This occurs during one-shot operation.

The continuous/one-shot block (Figure 14-71) controls the start/stop and reset (zero) functions of the Mod4 counter via a mono-shot type of action that can be triggered by the stop-value comparator and re-armed via software control.

Once armed, the eCAP module waits for 1-4 (defined by stop-value) capture events before freezing both the Mod4 counter and contents of CAP1-4 registers (time-stamps).

Re-arming prepares the eCAP module for another capture sequence. Also re-arming clears (to zero) the Mod4 counter and permits loading of CAP1-4 registers again, providing the CAPLDEN bit is set.

In continuous mode, the Mod4 counter continues to run (0->1->2->3->0, the one-shot action is ignored, and capture values continue to be written to CAP1-4 in a circular buffer sequence.

Figure 14-71. Continuous/One-shot Block Diagram

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.3.2.2.4 32-Bit Counter and Phase Control

This counter (Figure 14-72) provides the time-base for event captures, and is clocked via the system clock.

A phase register is provided to achieve synchronization with other counters, via a hardware and software forced sync. This is useful in APWM mode when a phase offset between modules is needed.

On any of the four event loads, an option to reset the 32-bit counter is given. This is useful for time difference capture. The 32-bit counter value is captured first, then it is reset to 0 by any of the LD1-LD4 signals.

Figure 14-72. Counter and Synchronization Block Diagram

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.3.2.2.5 CAP1-CAP4 Registers

These 32-bit registers are fed by the 32-bit counter timer bus, CTR[0-31] and are loaded (capture a time-stamp) when their respective LD inputs are strobed.

Loading of the capture registers can be inhibited via control bit CAPLDEN. During one-shot operation, this bit is cleared (loading is inhibited) automatically when a stop condition occurs, StopValue = Mod4.

CAP1 and CAP2 registers become the active period and compare registers, respectively, in APWM mode.

CAP3 and CAP4 registers become the respective shadow registers (APRD and ACMP) for CAP1 and CAP2 during APWM operation.

14.3.2.2.6 Interrupt Control

An Interrupt can be generated on capture events (CEVT1-CEVT4, CTROVF) or APWM events (CTR = PRD, CTR = CMP). See Figure 14-73.

A counter overflow event (FFFF FFFFh->0000 0000h) is also provided as an interrupt source (CTROVF).

The capture events are edge and sequencer qualified (that is, ordered in time) by the polarity select and Mod4 gating, respectively.

One of these events can be selected as the interrupt source (from the eCAP*n* module) going to the interrupt controller.

Seven interrupt events (CEVT1, CEVT2, CEVT3, CEVT4, CNTOVF, CTR = PRD, CTR = CMP) can be generated. The interrupt enable register (ECEINT) is used to enable/disable individual interrupt event sources. The interrupt flag register (ECFLG) indicates if any interrupt event has been latched and contains the global interrupt flag bit (INT). An interrupt pulse is generated to the interrupt controller only if any of the interrupt events are enabled, the flag bit is 1, and the INT flag bit is 0. The interrupt service routine must clear the global interrupt flag bit and the serviced event via the interrupt clear register (ECCLR) before any other interrupt pulses are generated. You can force an interrupt event via the interrupt force register (ECFRC). This is useful for test purposes.

14.3.2.2.7 Shadow Load and Lockout Control

In capture mode, this logic inhibits (locks out) any shadow loading of CAP1 or CAP2 from APRD and ACMP registers, respectively.

In APWM mode, shadow loading is active and two choices are permitted:

- Immediate - APRD or ACMP are transferred to CAP1 or CAP2 immediately upon writing a new value.
- On period equal, CTR[31:0] = PRD[31:0]

NOTE: The CEVT1, CEVT2, CEVT3, CEVT4 flags are only active in capture mode (ECCTL2[CAP/APWM == 0]). The CTR = PRD, CTR = CMP flags are only valid in APWM mode (ECCTL2[CAP/APWM == 1]). CNTOVF flag is valid in both modes.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 14-73. Interrupts in eCAP Module

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.3.2.2.8 APWM Mode Operation

Main operating highlights of the APWM section:

- The time-stamp counter bus is made available for comparison via 2 digital (32-bit) comparators.
- When CAP1/2 registers are not used in capture mode, their contents can be used as Period and Compare values in APWM mode.
- Double buffering is achieved via shadow registers APRD and ACMP (CAP3/4). The shadow register contents are transferred over to CAP1/2 registers either immediately upon a write, or on a CTR = PRD trigger.
- In APWM mode, writing to CAP1/CAP2 active registers will also write the same value to the corresponding shadow registers CAP3/CAP4. This emulates immediate mode. Writing to the shadow registers CAP3/CAP4 will invoke the shadow mode.
- During initialization, you must write to the active registers for both period and compare. This automatically copies the initial values into the shadow values. For subsequent compare updates, during run-time, you only need to use the shadow registers.

Figure 14-74. PWM Waveform Details Of APWM Mode Operation

The behavior of APWM active-high mode (APWMPOL == 0) is:

```
CMP = 0x00000000, output low for duration of period (0% duty)
CMP = 0x00000001, output high 1 cycle
CMP = 0x00000002, output high 2 cycles
CMP = PERIOD, output high except for 1 cycle (<100% duty)
CMP = PERIOD+1, output high for complete period (100% duty)
CMP > PERIOD+1, output high for complete period
```

The behavior of APWM active-low mode (APWMPOL == 1) is:

```
CMP = 0x00000000, output high for duration of period (0% duty)
CMP = 0x00000001, output low 1 cycle
CMP = 0x00000002, output low 2 cycles
CMP = PERIOD, output low except for 1 cycle (<100% duty)
CMP = PERIOD+1, output low for complete period (100% duty)
```


Copyright © 2011, Texas Instruments Incorporated

```
CMP > PERIOD+1, output low for complete period
```

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.3.3 Use Cases

The following sections will provide Applications examples and code snippets to show how to configure and operate the eCAP module. For clarity and ease of use, below are useful #defines which will help in the understanding of the examples.

```
// ECCTL1 ( ECAP Control Reg 1)
//==========================
// CAPxPOL bits
#define    EC_RISING              0x0
#define    EC_FALLING             0x1

// CTRRSTx bits
#define    EC_ABS_MODE            0x0
#define    EC_DELTA_MODE          0x1

// PRESCALE bits
#define    EC_BYPASS              0x0
#define    EC_DIV1                0x0
#define    EC_DIV2                0x1
#define    EC_DIV4                0x2
#define    EC_DIV6                0x3
#define    EC_DIV8                0x4
#define    EC_DIV10               0x5

// ECCTL2 ( ECAP Control Reg 2)
//==========================
// CONT/ONESHOT bit
#define    EC_CONTINUOUS          0x0
#define    EC_ONESHOT             0x1

// STOPVALUE bit
#define    EC_EVENT1              0x0
#define    EC_EVENT2              0x1
#define    EC_EVENT3              0x2
#define    EC_EVENT4              0x3

// RE-ARM bit
#define    EC_ARM                 0x1

// TSCTRSTOP bit
#define    EC_FREEZE              0x0
#define    EC_RUN                 0x1

// SYNCO_SEL bit
#define    EC_SYNCIN              0x0
#define    EC_CTR_PRD             0x1
#define    EC_SYNCO_DIS           0x2

// CAP/APWM mode bit
#define    EC_CAP_MODE            0x0
#define    EC_APWM_MODE           0x1

// APWMPOL bit
#define    EC_ACTV_HI             0x0
#define    EC_ACTV_LO             0x1

// Generic
#define    EC_DISABLE             0x0
#define    EC_ENABLE              0x1
#define    EC_FORCE               0x1
```

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.3.3.1 Absolute Time-Stamp Operation Rising Edge Trigger Example

Figure 14-75 shows an example of continuous capture operation (Mod4 counter wraps around). In this figure, TSCTR counts-up without resetting and capture events are qualified on the rising edge only, this gives period (and frequency) information.

On an event, the TSCTR contents (time-stamp) is first captured, then Mod4 counter is incremented to the next state. When the TSCTR reaches FFFF FFFFh (maximum value), it wraps around to 0000 0000h (not shown in Figure 14-75), if this occurs, the CTROVF (counter overflow) flag is set, and an interrupt (if enabled) occurs, CTROVF (counter overflow) Flag is set, and an Interrupt (if enabled) occurs. Captured time-stamps are valid at the point indicated by the diagram, after the 4th event, hence event CEVT4 can conveniently be used to trigger an interrupt and the CPU can read data from the CAP*n* registers.

Figure 14-75. Capture Sequence for Absolute Time-Stamp, Rising Edge Detect

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-59. ECAP Initialization for CAP Mode Absolute Time, Rising Edge Trigger

Register	Bit	Value
ECCTL1	CAP1POL	EC_RISING
ECCTL1	CAP2POL	EC_RISING
ECCTL1	CAP3POL	EC_RISING
ECCTL1	CAP4POL	EC_RISING
ECCTL1	CTRRST1	EC_ABS_MODE
ECCTL1	CTRRST2	EC_ABS_MODE
ECCTL1	CTRRST3	EC_ABS_MODE
ECCTL1	CTRRST4	EC_ABS_MODE
ECCTL1	CAPLDEN	EC_ENABLE
ECCTL1	PRESCALE	EC_DIV1
ECCTL2	CAP_APWM	EC_CAP_MODE
ECCTL2	CONT_ONESHT	EC_CONTINUOUS
ECCTL2	SYNCO_SEL	EC_SYNCO_DIS
ECCTL2	SYNCI_EN	EC_DISABLE
ECCTL2	TSCTRSTOP	EC_RUN

Example 14-18. Code Snippet for CAP Mode Absolute Time, Rising Edge Trigger

```
// Code snippet for CAP mode Absolute Time, Rising edge trigger

// Run Time ( e.g. CEVT4 triggered ISR call)
//=========================================
    TSt1 = ECAPxRegs.CAP1;        // Fetch Time-Stamp captured at t1
    TSt2 = ECAPxRegs.CAP2;        // Fetch Time-Stamp captured at t2
    TSt3 = ECAPxRegs.CAP3;        // Fetch Time-Stamp captured at t3
    TSt4 = ECAPxRegs.CAP4;        // Fetch Time-Stamp captured at t4

    Period1 = TSt2-TSt1;        // Calculate 1st period
    Period2 = TSt3-TSt2;        // Calculate 2nd period
    Period3 = TSt4-TSt3;        // Calculate 3rd period
```

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.3.3.2 Absolute Time-Stamp Operation Rising and Falling Edge Trigger Example

In Figure 14-76 the eCAP operating mode is almost the same as in the previous section except capture events are qualified as either rising or falling edge, this now gives both period and duty cycle information: Period1 = $t_3 – t_1$, Period2 = $t_5 – t_3$, …etc. Duty Cycle1 (on-time %) = $(t_2 – t_1)$ / Period1 x 100%, etc. Duty Cycle1 (off-time %) = $(t_3 – t_2)$ / Period1 x 100%, etc.

Figure 14-76. Capture Sequence for Absolute Time-Stamp, Rising and Falling Edge Detect

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-60. ECAP Initialization for CAP Mode Absolute Time, Rising and Falling Edge Trigger

Register	Bit	Value
ECCTL1	CAP1POL	EC_RISING
ECCTL1	CAP2POL	EC_FALLING
ECCTL1	CAP3POL	EC_RISING
ECCTL1	CAP4POL	EC_FALLING
ECCTL1	CTRRST1	EC_ABS_MODE
ECCTL1	CTRRST2	EC_ABS_MODE
ECCTL1	CTRRST3	EC_ABS_MODE
ECCTL1	CTRRST4	EC_ABS_MODE
ECCTL1	CAPLDEN	EC_ENABLE
ECCTL1	PRESCALE	EC_DIV1
ECCTL2	CAP_APWM	EC_CAP_MODE
ECCTL2	CONT_ONESHT	EC_CONTINUOUS
ECCTL2	SYNCO_SEL	EC_SYNCO_DIS
ECCTL2	SYNCI_EN	EC_DISABLE
ECCTL2	TSCTRSTOP	EC_RUN

Example 14-19. Code Snippet for CAP Mode Absolute Time, Rising and Falling Edge Trigger

```
// Code snippet for CAP mode Absolute Time, Rising & Falling edge triggers

// Run Time ( e.g. CEVT4 triggered ISR call)
//=========================================
    TSt1 = ECAPxRegs.CAP1;        // Fetch Time-Stamp captured at t1
    TSt2 = ECAPxRegs.CAP2;        // Fetch Time-Stamp captured at t2
    TSt3 = ECAPxRegs.CAP3;        // Fetch Time-Stamp captured at t3
    TSt4 = ECAPxRegs.CAP4;        // Fetch Time-Stamp captured at t4

    Period1 = TSt3-TSt1;        // Calculate 1st period
    DutyOnTime1 = TSt2-TSt1;    // Calculate On time
    DutyOffTime1 = TSt3-TSt2;    // Calculate Off time
```

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.3.3.3 Time Difference (Delta) Operation Rising Edge Trigger Example

Figure 14-77 shows how the eCAP module can be used to collect Delta timing data from pulse train waveforms. Here Continuous Capture mode (TSCTR counts-up without resetting, and Mod4 counter wraps around) is used. In Delta-time mode, TSCTR is Reset back to Zero on every valid event. Here Capture events are qualified as Rising edge only. On an event, TSCTR contents (time-stamp) is captured first, and then TSCTR is reset to Zero. The Mod4 counter then increments to the next state. If TSCTR reaches FFFF FFFFh (maximum value), before the next event, it wraps around to 0000 0000h and continues, a CNTOVF (counter overflow) Flag is set, and an Interrupt (if enabled) occurs. The advantage of Delta-time Mode is that the CAP*n* contents directly give timing data without the need for CPU calculations: Period1 = T_1, Period2 = T_2,…etc. As shown in Figure 14-77, the CEVT1 event is a good trigger point to read the timing data, T_1, T_2, T_3, T_4 are all valid here.

Figure 14-77. Capture Sequence for Delta Mode Time-Stamp, Rising Edge Detect

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-61. ECAP Initialization for CAP Mode Delta Time, Rising Edge Trigger

Register	Bit	Value
ECCTL1	CAP1POL	EC_RISING
ECCTL1	CAP2POL	EC_RISING
ECCTL1	CAP3POL	EC_RISING
ECCTL1	CAP4POL	EC_RISING
ECCTL1	CTRRST1	EC_DELTA_MODE
ECCTL1	CTRRST2	EC_DELTA_MODE
ECCTL1	CTRRST3	EC_DELTA_MODE
ECCTL1	CTRRST4	EC_DELTA_MODE
ECCTL1	CAPLDEN	EC_ENABLE
ECCTL1	PRESCALE	EC_DIV1
ECCTL2	CAP_APWM	EC_CAP_MODE
ECCTL2	CONT_ONESHT	EC_CONTINUOUS
ECCTL2	SYNCO_SEL	EC_SYNCO_DIS
ECCTL2	SYNCI_EN	EC_DISABLE
ECCTL2	TSCTRSTOP	EC_RUN

Example 14-20. Code Snippet for CAP Mode Delta Time, Rising Edge Trigger

```
// Code snippet for CAP mode Delta Time, Rising edge trigger

// Run Time ( e.g. CEVT1 triggered ISR call)
//=========================================
// Note: here Time-stamp directly represents the Period value.
    Period4 = ECAPxRegs.CAP1;    // Fetch Time-Stamp captured at T1
    Period1 = ECAPxRegs.CAP2;    // Fetch Time-Stamp captured at T2
    Period2 = ECAPxRegs.CAP3;    // Fetch Time-Stamp captured at T3
    Period3 = ECAPxRegs.CAP4;    // Fetch Time-Stamp captured at T4
```

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.3.3.4 Time Difference (Delta) Operation Rising and Falling Edge Trigger Example

In Figure 14-78 the eCAP operating mode is almost the same as in previous section except Capture events are qualified as either Rising or Falling edge, this now gives both Period and Duty cycle information: Period1 = $T_1 + T_2$, Period2 = $T_3 + T_4$, …etc Duty Cycle1 (on-time %) = T_1 / Period1 × 100%, etc Duty Cycle1 (off-time %) = T_2 / Period1 × 100%, etc

During initialization, you must write to the active registers for both period and compare. This will then automatically copy the init values into the shadow values. For subsequent compare updates, that is, during run-time, only the shadow registers must be used.

Figure 14-78. Capture Sequence for Delta Mode Time-Stamp, Rising and Falling Edge Detect

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-62. ECAP Initialization for CAP Mode Delta Time, Rising and Falling Edge Triggers

Register	Bit	Value
ECCTL1	CAP1POL	EC_RISING
ECCTL1	CAP2POL	EC_FALLING
ECCTL1	CAP3POL	EC_RISING
ECCTL1	CAP4POL	EC_FALLING
ECCTL1	CTRRST1	EC_DELTA_MODE
ECCTL1	CTRRST2	EC_DELTA_MODE
ECCTL1	CTRRST3	EC_DELTA_MODE
ECCTL1	CTRRST4	EC_DELTA_MODE
ECCTL1	CAPLDEN	EC_ENABLE
ECCTL1	PRESCALE	EC_DIV1
ECCTL2	CAP_APWM	EC_CAP_MODE
ECCTL2	CONT_ONESHT	EC_CONTINUOUS
ECCTL2	SYNCO_SEL	EC_SYNCO_DIS
ECCTL2	SYNCI_EN	EC_DISABLE
ECCTL2	TSCTRSTOP	EC_RUN

Example 14-21. Code Snippet for CAP Mode Delta Time, Rising and Falling Edge Triggers

```
// Code snippet for CAP mode Delta Time, Rising and Falling edge triggers

// Run Time ( e.g. CEVT1 triggered ISR call)
//=========================================
// Note: here Time-stamp directly represents the Duty cycle values.
    DutyOnTime1 =  ECAPxRegs.CAP2;    // Fetch Time-Stamp captured at T2
    DutyOffTime1 = ECAPxRegs.CAP3;    // Fetch Time-Stamp captured at T3
    DutyOnTime2 =  ECAPxRegs.CAP4;    // Fetch Time-Stamp captured at T4
    DutyOffTime2 = ECAPxRegs.CAP1;    // Fetch Time-Stamp captured at T1

    Period1 = DutyOnTime1 + DutyOffTime1;
    Period2 = DutyOnTime2 + DutyOffTime2;
```

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.3.3.5 Application of the APWM Mode

14.3.3.5.1 Simple PWM Generation (Independent Channel/s) Example

In this example, the eCAP module is configured to operate as a PWM generator. Here a very simple single channel PWM waveform is generated from output pin APWM*n*. The PWM polarity is active high, which means that the compare value (CAP2 reg is now a compare register) represents the on-time (high level) of the period. Alternatively, if the APWMPOL bit is configured for active low, then the compare value represents the off-time.

Figure 14-79. PWM Waveform Details of APWM Mode Operation

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-63. ECAP Initialization for APWM Mode

Register	Bit	Value
CAP1	CAP1	0x1000
CTRPHS	CTRPHS	0x0
ECCTL2	CAP_APWM	EC_APWM_MODE
ECCTL2	APWMPOL	EC_ACTV_HI
ECCTL2	SYNCI_EN	EC_DISABLE
ECCTL2	SYNCO_SEL	EC_SYNCO_DIS
ECCTL2	TSCTRSTOP	EC_RUN

Example 14-22. Code Snippet for APWM Mode

```
// Code snippet for APWM mode Example 1

// Run Time (Instant 1, e.g. ISR call)
//=====================
   ECAPxRegs.CAP2 = 0x300;      // Set Duty cycle i.e. compare value

// Run Time (Instant 2, e.g. another ISR call)
//=====================
   ECAPxRegs.CAP2 = 0x500;      // Set Duty cycle i.e. compare value
```

14.3.3.5.2 Multichannel PWM Generation with Synchronization Example

Figure 14-80 takes advantage of the synchronization feature between eCAP modules. Here 4 independent PWM channels are required with different frequencies, but at integer multiples of each other to avoid "beat" frequencies. Hence one eCAP module is configured as the Master and the remaining 3 are Slaves all receiving their synch pulse (CTR = PRD) from the master. Note the Master is chosen to have the lower frequency (F1 = 1/20,000) requirement. Here Slave2 Freq = 2 × F1, Slave3 Freq = 4 × F1 and Slave4 Freq = 5 × F1. Note here values are in decimal notation. Also, only the APWM1 output waveform is shown.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 14-80. Multichannel PWM Example Using 4 eCAP Modules

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-64. ECAP1 Initialization for Multichannel PWM Generation with Synchronization

Register	Bit	Value
CAP1	CAP1	20000
CTRPHS	CTRPHS	0
ECCTL2	CAP_APWM	EC_APWM_MODE
ECCTL2	APWMPOL	EC_ACTV_HI
ECCTL2	SYNCI_EN	EC_DISABLE
ECCTL2	SYNCO_SEL	EC_CTR_PRD
ECCTL2	TSCTRSTOP	EC_RUN

Table 14-65. ECAP2 Initialization for Multichannel PWM Generation with Synchronization

Register	Bit	Value
CAP1	CAP1	10000
CTRPHS	CTRPHS	0
ECCTL2	CAP_APWM	EC_APWM_MODE
ECCTL2	APWMPOL	EC_ACTV_HI
ECCTL2	SYNCI_EN	EC_ENABLE
ECCTL2	SYNCO_SEL	EC_SYNCI
ECCTL2	TSCTRSTOP	EC_RUN

Table 14-66. ECAP3 Initialization for Multichannel PWM Generation with Synchronization

Register	Bit	Value
CAP1	CAP1	5000
CTRPHS	CTRPHS	0
ECCTL2	CAP_APWM	EC_APWM_MODE
ECCTL2	APWMPOL	EC_ACTV_HI
ECCTL2	SYNCI_EN	EC_ENABLE
ECCTL2	SYNCO_SEL	EC_SYNCI
ECCTL2	TSCTRSTOP	EC_RUN

Table 14-67. ECAP4 Initialization for Multichannel PWM Generation with Synchronization

Register	Bit	Value
CAP1	CAP1	4000
CTRPHS	CTRPHS	0
ECCTL2	CAP_APWM	EC_APWM_MODE
ECCTL2	APWMPOL	EC_ACTV_HI
ECCTL2	SYNCI_EN	EC_ENABLE
ECCTL2	SYNCO_SEL	EC_SYNCO_DIS
ECCTL2	TSCTRSTOP	EC_RUN

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Example 14-23. Code Snippet for Multichannel PWM Generation with Synchronization

```
// Code snippet for APWM mode Example 2

// Run Time (Note: Example execution of one run-time instant)
//==========================================================
   ECAP1Regs.CAP2 = 7000;    // Set Duty cycle i.e., compare value = 7000
   ECAP2Regs.CAP2 = 2000;    // Set Duty cycle i.e., compare value = 2000
   ECAP3Regs.CAP2 = 550;     // Set Duty cycle i.e., compare value = 550
   ECAP4Regs.CAP2 = 6500;    // Set Duty cycle i.e., compare value = 6500
```

14.3.3.5.3 Multichannel PWM Generation with Phase Control Example

In Figure 14-81, the Phase control feature of the APWM mode is used to control a 3 phase Interleaved DC/DC converter topology. This topology requires each phase to be off-set by 120° from each other. Hence if “Leg” 1 (controlled by APWM1) is the reference Leg (or phase), that is, 0°, then Leg 2 need 120° off-set and Leg 3 needs 240° off-set. The waveforms in Figure 14-81 show the timing relationship between each of the phases (Legs). Note eCAP1 module is the Master and issues a sync out pulse to the slaves (modules 2, 3) whenever TSCTR = Period value.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 14-81. Multiphase (channel) Interleaved PWM Example Using 3 eCAP Modules

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-68. ECAP1 Initialization for Multichannel PWM Generation with Phase Control

Register	Bit	Value
CAP1	CAP1	1200
CTRPHS	CTRPHS	0
ECCTL2	CAP_APWM	EC_APWM_MODE
ECCTL2	APWMPOL	EC_ACTV_HI
ECCTL2	SYNCI_EN	EC_DISABLE
ECCTL2	SYNCO_SEL	EC_CTR_PRD
ECCTL2	TSCTRSTOP	EC_RUN

Table 14-69. ECAP2 Initialization for Multichannel PWM Generation with Phase Control

Register	Bit	Value
CAP1	CAP1	1200
CTRPHS	CTRPHS	800
ECCTL2	CAP_APWM	EC_APWM_MODE
ECCTL2	APWMPOL	EC_ACTV_HI
ECCTL2	SYNCI_EN	EC_ENABLE
ECCTL2	SYNCO_SEL	EC_SYNCI
ECCTL2	TSCTRSTOP	EC_RUN

Table 14-70. ECAP3 Initialization for Multichannel PWM Generation with Phase Control

Register	Bit	Value
CAP1	CAP1	1200
CTRPHS	CTRPHS	400
ECCTL2	CAP_APWM	EC_APWM_MODE
ECCTL2	APWMPOL	EC_ACTV_HI
ECCTL2	SYNCI_EN	EC_ENABLE
ECCTL2	SYNCO_SEL	EC_SYNCO_DIS
ECCTL2	TSCTRSTOP	EC_RUN

Example 14-24. Code Snippet for Multichannel PWM Generation with Phase Control

```
// Code snippet for APWM mode Example 3


// Run Time (Note: Example execution of one run-time instant)
//=========================================================
// All phases are set to the same duty cycle
  ECAP1Regs.CAP2 = 700;     // Set Duty cycle i.e. compare value = 700
  ECAP2Regs.CAP2 = 700;     // Set Duty cycle i.e. compare value = 700
  ECAP3Regs.CAP2 = 700;     // Set Duty cycle i.e. compare value = 700
```

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.3.4 eCAP Module Control and Status Registers

Table 14-71 shows the eCAP module control and status register set. All 32-bit registers are aligned on even address boundaries and are organized in little-endian mode. The 16 least-significant bits of a 32-bit register are located on lowest address (even address).

NOTE: In APWM mode, writing to CAP1/CAP2 active registers also writes the same value to the corresponding shadow registers CAP3/CAP4. This emulates immediate mode. Writing to the shadow registers CAP3/CAP4 invokes the shadow mode.

Table 14-71. Control and Status Register Set

Offset	Acronym	Description	Size (×16)	Section
0h	TSCTR	Time-Stamp Counter Register	2	Section 14.3.4.1
4h	CTRPHS	Counter Phase Offset Value Register	2	Section 14.3.4.2
8h	CAP1	Capture 1 Register	2	Section 14.3.4.3
Ch	CAP2	Capture 2 Register	2	Section 14.3.4.4
10h	CAP3	Capture 3 Register	2	Section 14.3.4.5
14h	CAP4	Capture 4 Register	2	Section 14.3.4.6
28h	ECCTL1	Capture Control Register 1	1	Section 14.3.4.7
2Ah	ECCTL2	Capture Control Register 2	1	Section 14.3.4.8
2Ch	ECEINT	Capture Interrupt Enable Register	1	Section 14.3.4.9
2Eh	ECFLG	Capture Interrupt Flag Register	1	Section 14.3.4.10
30h	ECCLR	Capture Interrupt Clear Register	1	Section 14.3.4.11
32h	ECFRC	Capture Interrupt Force Register	1	Section 14.3.4.12
5Ch	REVID	Revision ID Register	2	Section 14.4.3.25

14.3.4.1 Time-Stamp Counter Register (TSCTR)

The time-stamp counter register (TSCTR) is shown in Figure 14-82 and described in Table 14-72.

Figure 14-82. Time-Stamp Counter Register (TSCTR)

31	0
TSCTR	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 14-72. Time-Stamp Counter Register (TSCTR) Field Descriptions

Bit	Field	Value	Description
31-0	TSCTR	0-FFFF FFFFh	Active 32-bit counter register that is used as the capture time-base

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.3.4.2 Counter Phase Control Register (CTRPHS)

The counter phase control register (CTRPHS) is shown in Figure 14-83 and described in Table 14-73.

Figure 14-83. Counter Phase Control Register (CTRPHS)

31	0
CTRPHS	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 14-73. Counter Phase Control Register (CTRPHS) Field Descriptions

Bit	Field	Value	Description
31-0	CTRPHS	0-FFFF FFFFh	Counter phase value register that can be programmed for phase lag/lead. This register shadows TSCTR and is loaded into TSCTR upon either a SYNCI event or S/W force via a control bit. Used to achieve phase control synchronization with respect to other eCAP and EPWM time-bases.

14.3.4.3 Capture 1 Register (CAP1)

The capture 1 register (CAP1) is shown in Figure 14-84 and described in Table 14-74.

Figure 14-84. Capture 1 Register (CAP1)

31	0
CAP1	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 14-74. Capture 1 Register (CAP1) Field Descriptions

Bit	Field	Value	Description
31-0	CAP1	0-FFFF FFFFh	This register can be loaded (written) by: • Time-Stamp (i.e., counter value) during a capture event • Software - may be useful for test purposes • APRD active register when used in APWM mode

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.3.4.4 Capture 2 Register (CAP2)

The capture 2 register (CAP2) is shown in Figure 14-85 and described in Table 14-75.

Figure 14-85. Capture 2 Register (CAP2)

31	0
CAP2	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 14-75. Capture 2 Register (CAP2) Field Descriptions

Bit	Field	Value	Description
31-0	CAP2	0-FFFF FFFFh	This register can be loaded (written) by: • Time-Stamp (i.e., counter value) during a capture event • Software - may be useful for test purposes • ACMP active register when used in APWM mode

14.3.4.5 Capture 3 Register (CAP3)

The capture 3 register (CAP3) is shown in Figure 14-86 and described in Table 14-76.

Figure 14-86. Capture 3 Register (CAP3)

31	0
CAP3	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 14-76. Capture 3 Register (CAP3) Field Descriptions

Bit	Field	Value	Description
31-0	CAP3	0-FFFF FFFFh	In CMP mode, this is a time-stamp capture register. In APWM mode, this is the period shadow (APRD) register. You update the PWM period value through this register. In this mode, CAP3 shadows CAP1.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.3.4.6 Capture 4 Register (CAP4)

The capture 4 register (CAP4) is shown in Figure 14-87 and described in Table 14-77.

Figure 14-87. Capture 4 Register (CAP4)

31	0
CAP4	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 14-77. Capture 4 Register (CAP4) Field Descriptions

Bit	Field	Value	Description
31-0	CAP4	0-FFFF FFFFh	In CMP mode, this is a time-stamp capture register. In APWM mode, this is the compare shadow (ACMP) register. You update the PWM compare value through this register. In this mode, CAP4 shadows CAP2.

14.3.4.7 ECAP Control Register 1 (ECCTL1)

The ECAP control register 1 (ECCTL1) is shown in Figure 14-88 and described in Table 14-78.

Figure 14-88. ECAP Control Register 1 (ECCTL1)

15	14	13	9	8
FREE/SOFT		PRESCALE		CAPLDEN
R/W-0		R/W-0		R/W-0

7	6	5	4	3	2	1	0
CTRRST4	CAP4POL	CTRRST3	CAP3POL	CTRRST2	CAP2POL	CTRRST1	CAP1POL
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 14-78. ECAP Control Register 1 (ECCTL1) Field Descriptions

Bit	Field	Value	Description
15-14	FREE/SOFT	0-3h	Emulation Control
		0	TSCTR counter stops immediately on emulation suspend
		1h	TSCTR counter runs until = 0
		2h-3h	TSCTR counter is unaffected by emulation suspend (Run Free)
13-9	PRESCALE	0-1Fh	Event Filter prescale select
		0	Divide by 1 (i.e,. no prescale, by-pass the prescaler)
		1	Divide by 2
		2h	Divide by 4
		3h	Divide by 6
		4h	Divide by 8
		5h	Divide by 10
		...	
		1Eh	Divide by 60
		1Fh	Divide by 62
8	CAPLDEN		Enable Loading of CAP1-4 registers on a capture event
		0	Disable CAP1-4 register loads at capture event time.
		1	Enable CAP1-4 register loads at capture event time.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-78. ECAP Control Register 1 (ECCTL1) Field Descriptions (continued)

Bit	Field	Value	Description
7	CTRRST4		Counter Reset on Capture Event 4
		0	*Do not* reset counter on Capture Event 4 (absolute time stamp operation)
		1	Reset counter after Capture Event 4 time-stamp has been captured (used in difference mode operation)
6	CAP4POL		Capture Event 4 Polarity select
		0	Capture Event 4 triggered on a rising edge (RE)
		1	Capture Event 4 triggered on a falling edge (FE)
5	CTRRST3		Counter Reset on Capture Event 3
		0	*Do not* reset counter on Capture Event 3 (absolute time stamp)
		1	Reset counter after Event 3 time-stamp has been captured (used in difference mode operation)
4	CAP3POL		Capture Event 3 Polarity select
		0	Capture Event 3 triggered on a rising edge (RE)
		1	Capture Event 3 triggered on a falling edge (FE)
3	CTRRST2		Counter Reset on Capture Event 2
		0	*Do not* reset counter on Capture Event 2 (absolute time stamp)
		1	Reset counter after Event 2 time-stamp has been captured (used in difference mode operation)
2	CAP2POL		Capture Event 2 Polarity select
		0	Capture Event 2 triggered on a rising edge (RE)
		1	Capture Event 2 triggered on a falling edge (FE)
1	CTRRST1		Counter Reset on Capture Event 1
		0	*Do not* reset counter on Capture Event 1 (absolute time stamp)
		1	Reset counter after Event 1 time-stamp has been captured (used in difference mode operation)
0	CAP1POL		Capture Event 1 Polarity select
		0	Capture Event 1 triggered on a rising edge (RE)
		1	Capture Event 1 triggered on a falling edge (FE)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.3.4.8 ECAP Control Register 2 (ECCTL2)

The ECAP control register 2 (ECCTL2) is shown in Figure 14-89 and described in Table 14-79.

Figure 14-89. ECAP Control Register 2 (ECCTL2)

15				11	10	9	8
Reserved					APWMPOL	CAP/APWM	SWSYNC
R-0					R/W-0	R/W-0	R/W-0

7	6	5	4	3	2	1	0
SYNCO_SEL		SYNCI_EN	TSCTRSTOP	REARM	STOP_WRAP		CONT/ONESHT
R/W-0		R/W-0	R/W-0	R/W-0	R/W-1		R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 14-79. ECAP Control Register 2 (ECCTL2) Field Descriptions

Bit	Field	Value	Description
15-11	Reserved	0	Reserved
10	APWMPOL		APWM output polarity select. This is applicable only in APWM operating mode
		0	Output is active high (Compare value defines high time)
		1	Output is active low (Compare value defines low time)
9	CAP/APWM		CAP/APWM operating mode select
		0	ECAP module operates in capture mode. This mode forces the following configuration: • Inhibits TSCTR resets via CTR = PRD event • Inhibits shadow loads on CAP1 and 2 registers • Permits user to enable CAP1-4 register load • ECAP*n*/APWM*n* pin operates as a capture input
		1	ECAP module operates in APWM mode. This mode forces the following configuration: • Resets TSCTR on CTR = PRD event (period boundary • Permits shadow loading on CAP1 and 2 registers • Disables loading of time-stamps into CAP1-4 registers • ECAP*n*/APWM*n* pin operates as a APWM output
8	SWSYNC		Software-forced Counter (TSCTR) Synchronizing. This provides a convenient software method to synchronize some or all ECAP time bases. In APWM mode, the synchronizing can also be done via the CTR = PRD event.
		0	Writing a zero has no effect. Reading always returns a zero
		1	Writing a one forces a TSCTR shadow load of current ECAP module and any ECAP modules down-stream providing the SYNCO_SEL bits are 0,0. After writing a 1, this bit returns to a zero. Note: Selection CTR = PRD is meaningful only in APWM mode; however, you can choose it in CAP mode if you find doing so useful.
7-6	SYNCO_SEL	0-3h	Sync-Out Select
		0	Select sync-in event to be the sync-out signal (pass through)
		1h	Select CTR = PRD event to be the sync-out signal
		2h	Disable sync out signal
		3h	Disable sync out signal
5	SYNCI_EN		Counter (TSCTR) Sync-In select mode
		0	Disable sync-in option
		1	Enable counter (TSCTR) to be loaded from CTRPHS register upon either a SYNCI signal or a S/W force event.
4	TSCTRSTOP		Time Stamp (TSCTR) Counter Stop (freeze) Control
		0	TSCTR stopped
		1	TSCTR free-running

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-79. ECAP Control Register 2 (ECCTL2) Field Descriptions (continued)

Bit	Field	Value	Description
3	RE-ARM		One-Shot Re-Arming Control, that is, wait for stop trigger. Note: The re-arm function is valid in one shot or continuous mode.
		0	Has no effect (reading always returns a 0)
		1	Arms the one-shot sequence as follows: 1) Resets the Mod4 counter to zero 2) Unfreezes the Mod4 counter 3) Enables capture register loads
2-1	STOP_WRAP	0-3h	Stop value for one-shot mode. This is the number (between 1-4) of captures allowed to occur before the CAP(1-4) registers are frozen, that is, capture sequence is stopped. Wrap value for continuous mode. This is the number (between 1-4) of the capture register in which the circular buffer wraps around and starts again.
		0	Stop after Capture Event 1 in one-shot mode. Wrap after Capture Event 1 in continuous mode.
		1h	Stop after Capture Event 2 in one-shot mode. Wrap after Capture Event 2 in continuous mode.
		2h	Stop after Capture Event 3 in one-shot mode. Wrap after Capture Event 3 in continuous mode.
		3h	Stop after Capture Event 4 in one-shot mode. Wrap after Capture Event 4 in continuous mode.
			Notes: STOP_WRAP is compared to Mod4 counter and, when equal, 2 actions occur: • Mod4 counter is stopped (frozen) • Capture register loads are inhibited In one-shot mode, further interrupt events are blocked until re-armed.
0	CONT/ONESHT		Continuous or one-shot mode control (applicable only in capture mode)
		0	Operate in continuous mode
		1	Operate in one-shot mode

14.3.4.9 ECAP Interrupt Enable Register (ECEINT)

The ECAP interrupt enable register (ECEINT) is shown in Figure 14-90 and described in Table 14-80.

The interrupt enable bits (CEVT*n*) block any of the selected events from generating an interrupt. Events will still be latched into the flag bit (ECFLG register) and can be forced/cleared via the ECFRC/ECCLR registers.

The proper procedure for configuring peripheral modes and interrupts is:

1. Disable global interrupts
2. Stop eCAP counter
3. Disable eCAP interrupts
4. Configure peripheral registers
5. Clear spurious eCAP interrupt flags
6. Enable eCAP interrupts
7. Start eCAP counter
8. Enable global interrupts

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 14-90. ECAP Interrupt Enable Register (ECEINT)

15							8
Reserved							
R-0							
7	6	5	4	3	2	1	0
CTR=CMP	CTR=PRD	CTROVF	CEVT4	CEVT3	CEVT2	CETV1	Reserved
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 14-80. ECAP Interrupt Enable Register (ECEINT) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved
7	CTR=CMP		Counter Equal Compare Interrupt Enable
		0	Disable Compare Equal as an Interrupt source
		1	Enable Compare Equal as an Interrupt source
6	CTR=PRD		Counter Equal Period Interrupt Enable
		0	Disable Period Equal as an Interrupt source
		1	Enable Period Equal as an Interrupt source
5	CTROVF		Counter Overflow Interrupt Enable
		0	Disable counter Overflow as an Interrupt source
		1	Enable counter Overflow as an Interrupt source
4	CEVT4		Capture Event 4 Interrupt Enable
		0	Disable Capture Event 4 as an Interrupt source
		1	Enable Capture Event 4 as an Interrupt source
3	CEVT3		Capture Event 3 Interrupt Enable
		0	Disable Capture Event 3 as an Interrupt source
		1	Enable Capture Event 3 as an Interrupt source
2	CEVT2		Capture Event 2 Interrupt Enable
		0	Disable Capture Event 2 as an Interrupt source
		1	Enable Capture Event 2 as an Interrupt source
1	CEVT1		Capture Event 1 Interrupt Enable
		0	Disable Capture Event 1 as an Interrupt source
		1	Enable Capture Event 1 as an Interrupt source
0	Reserved	0	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.3.4.10 ECAP Interrupt Flag Register (ECFLG)

The ECAP interrupt flag register (ECFLG) is shown in Figure 14-91 and described in Table 14-81.

Figure 14-91. ECAP Interrupt Flag Register (ECFLG)

15							8
Reserved							
R-0							
7	6	5	4	3	2	1	0
CTR=CMP	CTR=PRD	CTROVF	CEVT4	CETV3	CEVT2	CETV1	INT
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 14-81. ECAP Interrupt Flag Register (ECFLG) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved
7	CTR=CMP		Compare Equal Compare Status Flag. This flag is only active in APWM mode.
		0	Indicates no event occurred
		1	Indicates the counter (TSCTR) reached the compare register value (ACMP)
6	CTR=PRD		Counter Equal Period Status Flag. This flag is only active in APWM mode.
		0	Indicates no event occurred
		1	Indicates the counter (TSCTR) reached the period register value (APRD) and was reset.
5	CTROVF		Counter Overflow Status Flag. This flag is active in CAP and APWM mode.
		0	Indicates no event occurred.
		1	Indicates the counter (TSCTR) has made the transition from 0xFFFFFFFF to 0x00000000
4	CEVT4		Capture Event 4 Status Flag This flag is only active in CAP mode.
		0	Indicates no event occurred
		1	Indicates the fourth event occurred at ECAP*n* pin
3	CEVT3		Capture Event 3 Status Flag. This flag is active only in CAP mode.
		0	Indicates no event occurred.
		1	Indicates the third event occurred at ECAP*n* pin.
2	CEVT2		Capture Event 2 Status Flag. This flag is only active in CAP mode.
		0	Indicates no event occurred.
		1	Indicates the second event occurred at ECAP*n* pin.
1	CEVT1		Capture Event 1 Status Flag. This flag is only active in CAP mode.
		0	Indicates no event occurred.
		1	Indicates the first event occurred at ECAP*n* pin.
0	INT		Global Interrupt Status Flag
		0	Indicates no interrupt generated.
		1	Indicates that an interrupt was generated.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.3.4.11 ECAP Interrupt Clear Register (ECCLR)

The ECAP interrupt clear register (ECCLR) is shown in Figure 14-92 and described in Table 14-82.

Figure 14-92. ECAP Interrupt Clear Register (ECCLR)

15							8
Reserved							
R-0							
7	6	5	4	3	2	1	0
CTR=CMP	CTR=PRD	CTROVF	CEVT4	CETV3	CETV2	CETV1	INT
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 14-82. ECAP Interrupt Clear Register (ECCLR) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved
7	CTR=CMP		Counter Equal Compare Status Flag
		0	Writing a 0 has no effect. Always reads back a 0
		1	Writing a 1 clears the CTR=CMP flag condition
6	CTR=PRD		Counter Equal Period Status Flag
		0	Writing a 0 has no effect. Always reads back a 0
		1	Writing a 1 clears the CTR=PRD flag condition
5	CTROVF		Counter Overflow Status Flag
		0	Writing a 0 has no effect. Always reads back a 0
		1	Writing a 1 clears the CTROVF flag condition
4	CEVT4		Capture Event 4 Status Flag
		0	Writing a 0 has no effect. Always reads back a 0.
		1	Writing a 1 clears the CEVT3 flag condition.
3	CEVT3		Capture Event 3 Status Flag
		0	Writing a 0 has no effect. Always reads back a 0.
		1	Writing a 1 clears the CEVT3 flag condition.
2	CEVT2		Capture Event 2 Status Flag
		0	Writing a 0 has no effect. Always reads back a 0.
		0	Writing a 1 clears the CEVT2 flag condition.
1	CEVT1		Capture Event 1 Status Flag
		0	Writing a 0 has no effect. Always reads back a 0.
		1	Writing a 1 clears the CEVT1 flag condition.
0	INT		Global Interrupt Clear Flag
		0	Writing a 0 has no effect. Always reads back a 0.
		1	Writing a 1 clears the INT flag and enable further interrupts to be generated if any of the event flags are set to 1.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.3.4.12 ECAP Interrupt Forcing Register (ECFRC)

The ECAP interrupt forcing register (ECFRC) is shown in Figure 14-93 and described in Table 14-83.

Figure 14-93. ECAP Interrupt Forcing Register (ECFRC)

15	14	13	12	11	10	9	8
Reserved							
R-0							

7	6	5	4	3	2	1	0
CTR=CMP	CTR=PRD	CTROVF	CEVT4	CETV3	CETV2	CETV1	Reserved
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 14-83. ECAP Interrupt Forcing Register (ECFRC) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved
7	CTR=CMP		Force Counter Equal Compare Interrupt
		0	No effect. Always reads back a 0.
		1	Writing a 1 sets the CTR=CMP flag bit.
6	CTR=PRD		Force Counter Equal Period Interrupt
		0	No effect. Always reads back a 0.
		1	Writing a 1 sets the CTR=PRD flag bit.
5	CTROVF		Force Counter Overflow
		0	No effect. Always reads back a 0.
		1	Writing a 1 to this bit sets the CTROVF flag bit.
4	CEVT4		Force Capture Event 4
		0	No effect. Always reads back a 0.
		1	Writing a 1 sets the CEVT4 flag bit
3	CEVT3		Force Capture Event 3
		0	No effect. Always reads back a 0.
		1	Writing a 1 sets the CEVT3 flag bit
2	CEVT2		Force Capture Event 2
		0	No effect. Always reads back a 0.
		1	Writing a 1 sets the CEVT2 flag bit.
1	CEVT1		Force Capture Event 1
		0	No effect. Always reads back a 0.
		1	Writing a 1 sets the CEVT1 flag bit.
0	Reserved	0	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.3.4.13 Revision ID Register (REVID)

The revision ID register (REVID) is shown in Figure 14-94 and described in Table 14-84.

Figure 14-94. Revision ID Register (REVID)

31	0
REV	
R-44D2 2100h	

LEGEND: R = Read only; -*n* = value after reset

Table 14-84. Revision ID Register (REVID) Field Descriptions

Bit	Field	Value	Description
31-0	REV	44D2 2100h	Revision ID.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.4 Enhanced Quadrature Encoder Pulse (eQEP) Module

14.4.1 Introduction

A single track of slots patterns the periphery of an incremental encoder disk, as shown in Figure 14-95. These slots create an alternating pattern of dark and light lines. The disk count is defined as the number of dark/light line pairs that occur per revolution (lines per revolution). As a rule, a second track is added to generate a signal that occurs once per revolution (index signal: QEPI), which can be used to indicate an absolute position. Encoder manufacturers identify the index pulse using different terms such as index, marker, home position, and zero reference.

Figure 14-95. Optical Encoder Disk

To derive direction information, the lines on the disk are read out by two different photo-elements that "look" at the disk pattern with a mechanical shift of 1/4 the pitch of a line pair between them. This shift is realized with a reticle or mask that restricts the view of the photo-element to the desired part of the disk lines. As the disk rotates, the two photo-elements generate signals that are shifted 90 degrees out of phase from each other. These are commonly called the quadrature QEPA and QEPB signals. The clockwise direction for most encoders is defined as the QEPA channel going positive before the QEPB channel and vise versa as shown in Figure 14-96.

The encoder wheel typically makes one revolution for every revolution of the motor or the wheel may be at a geared rotation ratio with respect to the motor. Therefore, the frequency of the digital signal coming from the QEPA and QEPB outputs varies proportionally with the velocity of the motor. For example, a 2000-line encoder directly coupled to a motor running at 5000 revolutions per minute (rpm) results in a frequency of 166.6 KHz, so by measuring the frequency of either the QEPA or QEPB output, the processor can determine the velocity of the motor.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 14-96. QEP Encoder Output Signal for Forward/Reverse Movement

Legend: N = lines per revolution

Quadrature encoders from different manufacturers come with two forms of index pulse (gated index pulse or ungated index pulse) as shown in Figure 14-97. A nonstandard form of index pulse is ungated. In the ungated configuration, the index edges are not necessarily coincident with A and B signals. The gated index pulse is aligned to any of the four quadrature edges and width of the index pulse and can be equal to a quarter, half, or full period of the quadrature signal.

Figure 14-97. Index Pulse Example

Copyright © 2011, Texas Instruments Incorporated

Some typical applications of shaft encoders include robotics and even computer input in the form of a mouse. Inside your mouse you can see where the mouse ball spins a pair of axles (a left/right, and an up/down axle). These axles are connected to optical shaft encoders that effectively tell the computer how fast and in what direction the mouse is moving.

General Issues: Estimating velocity from a digital position sensor is a cost-effective strategy in motor control. Two different first order approximations for velocity may be written as:

$$v(k) \approx \frac{x(k) - x(k-1)}{T} = \frac{\Delta X}{T} \quad (1)$$

$$v(k) \approx \frac{X}{t(k) - t(k-1)} = \frac{X}{\Delta T} \quad (2)$$

where

v(k): Velocity at time instant k

x(k): Position at time instant k

x(k-1): Position at time instant k - 1

T: Fixed unit time or inverse of velocity calculation rate

ΔX: Incremental position movement in unit time

t(k): Time instant "k"

t(k-1): Time instant "k - 1"

X: Fixed unit position

ΔT: Incremental time elapsed for unit position movement.

Equation 1 is the conventional approach to velocity estimation and it requires a time base to provide unit time event for velocity calculation. Unit time is basically the inverse of the velocity calculation rate.

The encoder count (position) is read once during each unit time event. The quantity [x(k) - x(k-1)] is formed by subtracting the previous reading from the current reading. Then the velocity estimate is computed by multiplying by the known constant 1/T (where T is the constant time between unit time events and is known in advance).

Estimation based on Equation 1 has an inherent accuracy limit directly related to the resolution of the position sensor and the unit time period T. For example, consider a 500-line per revolution quadrature encoder with a velocity calculation rate of 400 Hz. When used for position the quadrature encoder gives a four-fold increase in resolution, in this case, 2000 counts per revolution. The minimum rotation that can be detected is therefore 0.0005 revolutions, which gives a velocity resolution of 12 rpm when sampled at 400 Hz. While this resolution may be satisfactory at moderate or high speeds, for example, 1% error at 1200 rpm, it would clearly prove inadequate at low speeds. In fact, at speeds below 12 rpm, the speed estimate would erroneously be zero much of the time.

At low speed, Equation 2 provides a more accurate approach. It requires a position sensor that outputs a fixed interval pulse train, such as the aforementioned quadrature encoder. The width of each pulse is defined by motor speed for a given sensor resolution. Equation 2 can be used to calculate motor speed by measuring the elapsed time between successive quadrature pulse edges. However, this method suffers from the opposite limitation, as does Equation 1. A combination of relatively large motor speeds and high sensor resolution makes the time interval ΔT small, and thus more greatly influenced by the timer resolution. This can introduce considerable error into high-speed estimates.

For systems with a large speed range (that is, speed estimation is needed at both low and high speeds), one approach is to use Equation 2 at low speed and have the software switch over to Equation 1 when the motor speed rises above some specified threshold.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.4.2 Functional Description

This section provides the eQEP inputs and functional description.

NOTE: Multiple identical eQEP modules can be contained in a system. The number of modules is device-dependent and is based on target application needs. In this document, the letter x within a signal or module name is used to indicate a generic eQEP instance on a device.

14.4.2.1 EQEP Inputs

The eQEP inputs include two pins for quadrature-clock mode or direction-count mode, an index (or 0 marker), and a strobe input.

- QEPA/XCLK and QEPB/XDIR: These two pins can be used in quadrature-clock mode or direction-count mode.
  - Quadrature-clock Mode: The eQEP encoders provide two square wave signals (A and B) 90 electrical degrees out of phase whose phase relationship is used to determine the direction of rotation of the input shaft and number of eQEP pulses from the index position to derive the relative position information. For forward or clockwise rotation, QEPA signal leads QEPB signal and vice versa. The quadrature decoder uses these two inputs to generate quadrature-clock and direction signals.
  - Direction-count Mode: In direction-count mode, direction and clock signals are provided directly from the external source. Some position encoders have this type of output instead of quadrature output. The QEPA pin provides the clock input and the QEPB pin provides the direction input.
- QEPI: Index or Zero Marker: The eQEP encoder uses an index signal to assign an absolute start position from which position information is incrementally encoded using quadrature pulses. This pin is connected to the index output of the eQEP encoder to optionally reset the position counter for each revolution. This signal can be used to initialize or latch the position counter on the occurrence of a desired event on the index pin.
- QEPS: Strobe Input: This general-purpose strobe signal can initialize or latch the position counter on the occurrence of a desired event on the strobe pin. This signal is typically connected to a sensor or limit switch to notify that the motor has reached a defined position.

14.4.2.2 Functional Description

The eQEP peripheral contains the following major functional units (as shown in Figure 14-98):

- Programmable input qualification for each pin (part of the GPIO MUX)
- Quadrature decoder unit (QDU)
- Position counter and control unit for position measurement (PCCU)
- Quadrature edge-capture unit for low-speed measurement (QCAP)
- Unit time base for speed/frequency measurement (UTIME)
- Watchdog timer for detecting stalls (QWDOG)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 14-98. Functional Block Diagram of the eQEP Peripheral

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.4.2.3 Quadrature Decoder Unit (QDU)

Figure 14-99 shows a functional block diagram of the QDU.

Figure 14-99. Functional Block Diagram of Decoder Unit

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.4.2.3.1 Position Counter Input Modes

Clock and direction input to position counter is selected using the QSRC bit in the eQEP decoder control register (QDECCTL), based on interface input requirement as follows:

- Quadrature-count mode
- Direction-count mode
- UP-count mode
- DOWN-count mode

14.4.2.3.1.1 Quadrature Count Mode

The quadrature decoder generates the direction and clock to the position counter in quadrature count mode.

Direction Decoding— The direction decoding logic of the eQEP circuit determines which one of the sequences (QEPA, QEPB) is the leading sequence and accordingly updates the direction information in the QDF bit in the eQEP status register (QEPSTS). Table 14-85 and Figure 14-100 show the direction decoding logic in truth table and state machine form. Both edges of the QEPA and QEPB signals are sensed to generate count pulses for the position counter. Therefore, the frequency of the clock generated by the eQEP logic is four times that of each input sequence. Figure 14-101 shows the direction decoding and clock generation from the eQEP input signals.

Phase Error Flag— In normal operating conditions, quadrature inputs QEPA and QEPB will be 90 degrees out of phase. The phase error flag (PHE) is set in the QFLG register when edge transition is detected simultaneously on the QEPA and QEPB signals to optionally generate interrupts. State transitions marked by dashed lines in Figure 14-100 are invalid transitions that generate a phase error.

Count Multiplication— The eQEP position counter provides 4x times the resolution of an input clock by generating a quadrature-clock (QCLK) on the rising/falling edges of both eQEP input clocks (QEPA and QEPB) as shown in Figure 14-101.

Reverse Count— In normal quadrature count operation, QEPA input is fed to the QA input of the quadrature decoder and the QEPB input is fed to the QB input of the quadrature decoder. Reverse counting is enabled by setting the SWAP bit in the eQEP decoder control register (QDECCTL). This will swap the input to the quadrature decoder thereby reversing the counting direction.

Table 14-85. Quadrature Decoder Truth Table

Previous Edge	Present Edge	QDIR	QPOSCNT
QA↑	QB↑	UP	Increment
	QB↓	DOWN	Decrement
	QA↓	TOGGLE	Increment or Decrement
QA↓	QB↓	UP	Increment
	QB↑	DOWN	Decrement
	QA↑	TOGGLE	Increment or Decrement
QB↑	QA↑	DOWN	Increment
	QA↓	UP	Decrement
	QB↓	TOGGLE	Increment or Decrement
QB↓	QA↓	DOWN	Increment
	QA↑	UP	Decrement
	QB↑	TOGGLE	Increment or Decrement

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 14-100. Quadrature Decoder State Machine

Figure 14-101. Quadrature-clock and Direction Decoding

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.4.2.3.1.2 Direction-count Mode

Some position encoders provide direction and clock outputs, instead of quadrature outputs. In such cases, direction-count mode can be used. QEPA input will provide the clock for position counter and the QEPB input will have the direction information. The position counter is incremented on every rising edge of a QEPA input when the direction input is high and decremented when the direction input is low.

14.4.2.3.1.3 Up-Count Mode

The counter direction signal is hard-wired for up count and the position counter is used to measure the frequency of the QEPA input. Setting of the XCR bit in the eQEP decoder control register (QDECCTL) enables clock generation to the position counter on both edges of the QEPA input, thereby increasing the measurement resolution by 2× factor.

14.4.2.3.1.4 Down-Count Mode

The counter direction signal is hardwired for a down count and the position counter is used to measure the frequency of the QEPA input. Setting of the XCR bit in the eQEP decoder control register (QDECCTL) enables clock generation to the position counter on both edges of a QEPA input, thereby increasing the measurement resolution by 2× factor.

14.4.2.3.2 eQEP Input Polarity Selection

Each eQEP input can be inverted using the in the eQEP decoder control register (QDECCTL[8:5]) control bits. As an example, setting of the QIP bit in QDECCTL inverts the index input.

14.4.2.3.3 Position-Compare Sync Output

The eQEP peripheral includes a position-compare unit that is used to generate the position-compare sync signal on compare match between the position counter register (QPOSCNT) and the position-compare register (QPOSCMP). This sync signal can be output using an index pin or strobe pin of the EQEP peripheral.

Setting the SOEN bit in the eQEP decoder control register (QDECCTL) enables the position-compare sync output and the SPSEL bit in QDECCTL selects either an eQEP index pin or an eQEP strobe pin.

14.4.2.4 Position Counter and Control Unit (PCCU)

The position counter and control unit provides two configuration registers (QEPCTL and QPOSCTL) for setting up position counter operational modes, position counter initialization/latch modes and position-compare logic for sync signal generation.

14.4.2.4.1 Position Counter Operating Modes

Position counter data may be captured in different manners. In some systems, the position counter is accumulated continuously for multiple revolutions and the position counter value provides the position information with respect to the known reference. An example of this is the quadrature encoder mounted on the motor controlling the print head in the printer. Here the position counter is reset by moving the print head to the home position and then position counter provides absolute position information with respect to home position.

In other systems, the position counter is reset on every revolution using index pulse and position counter provides rotor angle with respect to index pulse position.

Position counter can be configured to operate in following four modes

- Position Counter Reset on Index Event
- Position Counter Reset on Maximum Position
- Position Counter Reset on the first Index Event
- Position Counter Reset on Unit Time Out Event (Frequency Measurement)

Copyright © 2011, Texas Instruments Incorporated

In all the above operating modes, position counter is reset to 0 on overflow and to QPOSMAX register value on underflow. Overflow occurs when the position counter counts up after QPOSMAX value. Underflow occurs when position counter counts down after "0". Interrupt flag is set to indicate overflow/underflow in QFLG register.

14.4.2.4.1.1 Position Counter Reset on Index Event (QEPCTL[PCRM] = 00)

If the index event occurs during the forward movement, then position counter is reset to 0 on the next eQEP clock. If the index event occurs during the reverse movement, then the position counter is reset to the value in the QPOSMAX register on the next eQEP clock.

First index marker is defined as the quadrature edge following the first index edge. The eQEP peripheral records the occurrence of the first index marker (QEPSTS[FIMF]) and direction on the first index event marker (QEPSTS[FIDF]) in QEPSTS registers, it also remembers the quadrature edge on the first index marker so that same relative quadrature transition is used for index event reset operation.

For example, if the first reset operation occurs on the falling edge of QEPB during the forward direction, then all the subsequent reset must be aligned with the falling edge of QEPB for the forward rotation and on the rising edge of QEPB for the reverse rotation as shown in Figure 14-102.

The position-counter value is latched to the QPOSILAT register and direction information is recorded in the QEPSTS[QDLF] bit on every index event marker. The position-counter error flag (QEPSTS[PCEF]) and error interrupt flag (QFLG[PCE]) are set if the latched value is not equal to 0 or QPOSMAX. The position-counter error flag (QEPSTS[PCEF]) is updated on every index event marker and an interrupt flag (QFLG[PCE]) will be set on error that can be cleared only through software.

The index event latch configuration QEPCTL[IEL] bits are ignored in this mode and position counter error flag/interrupt flag are generated only in index event reset mode.

Figure 14-102. Position Counter Reset by Index Pulse for 1000 Line Encoder (QPOSMAX = 3999 or F9Fh)

Copyright © 2011, Texas Instruments Incorporated

14.4.2.4.1.2 Position Counter Reset on Maximum Position (QEPCTL[PCRM]=01)

If the position counter is equal to QPOSMAX, then the position counter is reset to 0 on the next eQEP clock for forward movement and position counter overflow flag is set. If the position counter is equal to ZERO, then the position counter is reset to QPOSMAX on the next QEP clock for reverse movement and position counter underflow flag is set. Figure 14-103 shows the position counter reset operation in this mode.

First index marker is defined as the quadrature edge following the first index edge. The eQEP peripheral records the occurrence of the first index marker (QEPSTS[FIMF]) and direction on the first index event marker (QEPSTS[FIDF]) in the QEPSTS registers; it also remembers the quadrature edge on the first index marker so that the same relative quadrature transition is used for the software index marker (QEPCTL[IEL]=11).

Figure 14-103. Position Counter Underflow/Overflow (QPOSMAX = 4)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.4.2.4.1.3 Position Counter Reset on the First Index Event (QEPCTL[PCRM] = 10)

If the index event occurs during forward movement, then the position counter is reset to 0 on the next eQEP clock. If the index event occurs during the reverse movement, then the position counter is reset to the value in the QPOSMAX register on the next eQEP clock. Note that this is done only on the first occurrence and subsequently the position counter value is not reset on an index event; rather, it is reset based on maximum position as described in Section 14.4.2.4.1.2.

First index marker is defined as the quadrature edge following the first index edge. The eQEP peripheral records the occurrence of the first index marker (QEPSTS[FIMF]) and direction on the first index event marker (QEPSTS[FIDF]) in QEPSTS registers. It also remembers the quadrature edge on the first index marker so that same relative quadrature transition is used for software index marker (QEPCTL[IEL]=11).

14.4.2.4.1.4 Position Counter Reset on Unit Time out Event (QEPCTL[PCRM] = 11)

In this mode, the QPOSCNT value is latched to the QPOSLAT register and then the QPOSCNT is reset (to 0 or QPOSMAX, depending on the direction mode selected by QDECCTL[QSRC] bits on a unit time event). This is useful for frequency measurement.

14.4.2.4.2 Position Counter Latch

The eQEP index and strobe input can be configured to latch the position counter (QPOSCNT) into QPOSILAT and QPOSSLAT, respectively, on occurrence of a definite event on these pins.

14.4.2.4.2.1 Index Event Latch

In some applications, it may not be desirable to reset the position counter on every index event and instead it may be required to operate the position counter in full 32-bit mode (QEPCTL[PCRM] = 01 and QEPCTL[PCRM] = 10 modes).

In such cases, the eQEP position counter can be configured to latch on the following events and direction information is recorded in the QEPSTS[QDLF] bit on every index event marker.

- Latch on Rising edge (QEPCTL[IEL] = 01)
- Latch on Falling edge (QEPCTL[IEL] = 10)
- Latch on Index Event Marker (QEPCTL[IEL] = 11)

This is particularly useful as an error checking mechanism to check if the position counter accumulated the correct number of counts between index events. As an example, the 1000-line encoder must count 4000 times when moving in the same direction between the index events.

The index event latch interrupt flag (QFLG[IEL]) is set when the position counter is latched to the QPOSILAT register. The index event latch configuration bits (QEPCTZ[IEL]) are ignored when QEPCTL[PCRM] = 00.

Latch on Rising Edge (QEPCTL[IEL] = 01)— The position counter value (QPOSCNT) is latched to the QPOSILAT register on every rising edge of an index input.

Latch on Falling Edge (QEPCTL[IEL] = 10)— The position counter value (QPOSCNT) is latched to the QPOSILAT register on every falling edge of index input.

Latch on Index Event Marker/Software Index Marker (QEPCTL[IEL] = 11)— The first index marker is defined as the quadrature edge following the first index edge. The eQEP peripheral records the occurrence of the first index marker (QEPSTS[FIMF]) and direction on the first index event marker (QEPSTS[FIDF]) in the QEPSTS registers. It also remembers the quadrature edge on the first index marker so that same relative quadrature transition is used for latching the position counter (QEPCTL[IEL] = 11).

Figure 14-104 shows the position counter latch using an index event marker.

Copyright © 2011, Texas Instruments Incorporated

Figure 14-104. Software Index Marker for 1000-line Encoder (QEPCTL[IEL] = 1)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.4.2.4.2.2 Strobe Event Latch

The position-counter value is latched to the QPOSSLAT register on the rising edge of the strobe input by clearing the QEPCTL[SEL] bit.

If the QEPCTL[SEL] bit is set, then the position counter value is latched to the QPOSSLAT register on the rising edge of the strobe input for forward direction and on the falling edge of the strobe input for reverse direction as shown in Figure 14-105.

The strobe event latch interrupt flag (QFLG[SEL]) is set when the position counter is latched to the QPOSSLAT register.

Figure 14-105. Strobe Event Latch (QEPCTL[SEL] = 1)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.4.2.4.3 Position Counter Initialization

The position counter can be initialized using following events:

- Index event
- Strobe event
- Software initialization

Index Event Initialization (IEI)— The QEPI index input can be used to trigger the initialization of the position counter at the rising or falling edge of the index input.

If the QEPCTL[IEI] bits are 10, then the position counter (QPOSCNT) is initialized with a value in the QPOSINIT register on the rising edge of strobe input for forward direction and on the falling edge of strobe input for reverse direction.

The index event initialization interrupt flag (QFLG[IEI]) is set when the position counter is initialized with a value in the QPOSINIT register.

Strobe Event Initialization (SEI)— If the QEPCTL[SEI] bits are 10, then the position counter is initialized with a value in the QPOSINIT register on the rising edge of strobe input.

If the QEPCTL[SEL] bits are 11, then the position counter (QPOSCNT) is initialized with a value in the QPOSINIT register on the rising edge of strobe input for forward direction and on the falling edge of strobe input for reverse direction.

The strobe event initialization interrupt flag (QFLG[SEI]) is set when the position counter is initialized with a value in the QPOSINIT register.

Software Initialization (SWI)— The position counter can be initialized in software by writing a 1 to the QEPCTL[SWI] bit, which will automatically be cleared after initialization.

14.4.2.4.4 eQEP Position-compare Unit

The eQEP peripheral includes a position-compare unit that is used to generate a sync output and/or interrupt on a position-compare match. Figure 14-106 shows a diagram. The position-compare (QPOSCMP) register is shadowed and shadow mode can be enabled or disabled using the QPOSCTL[PSSHDW] bit. If the shadow mode is not enabled, the CPU writes directly to the active position compare register.

Figure 14-106. eQEP Position-compare Unit

In shadow mode, you can configure the position-compare unit (QPOSCTL[PCLOAD]) to load the shadow register value into the active register on the following events and to generate the position-compare ready (QFLG[PCR]) interrupt after loading.

- Load on compare match
- Load on position-counter zero event

Copyright © 2011, Texas Instruments Incorporated

The position-compare match (QFLG[PCM]) is set when the position-counter value (QPOSCNT) matches with the active position-compare register (QPOSCMP) and the position-compare sync output of the programmable pulse width is generated on compare match to trigger an external device.

For example, if QPOSCMP = 2, the position-compare unit generates a position-compare event on 1 to 2 transitions of the eQEP position counter for forward counting direction and on 3 to 2 transitions of the eQEP position counter for reverse counting direction (see Figure 14-107).

Figure 14-129 shows the layout of the eQEP Position-Compare Control Register (QPOSCTL) and Table 14-101 describes the QPOSCTL bit fields.

Figure 14-107. eQEP Position-compare Event Generation Points

The pulse stretcher logic in the position-compare unit generates a programmable position-compare sync pulse output on the position-compare match. In the event of a new position-compare match while a previous position-compare pulse is still active, then the pulse stretcher generates a pulse of specified duration from the new position-compare event as shown in Figure 14-108.

Figure 14-108. eQEP Position-compare Sync Output Pulse Stretcher

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.4.2.5 eQEP Edge Capture Unit

The eQEP peripheral includes an integrated edge capture unit to measure the elapsed time between the unit position events as shown in Figure 14-109. This feature is typically used for low speed measurement using the following equation:

$$v(k) = \frac{X}{t(k) - t(k-1)} = \frac{X}{\Delta T} \quad (3)$$

where,

- X - Unit position is defined by integer multiple of quadrature edges (see Figure 14-110)
- ΔT - Elapsed time between unit position events
- v(k) - Velocity at time instant "k"

The eQEP capture timer (QCTMR) runs from prescaled SYSCLKOUT and the prescaler is programmed by the QCAPCTL[CCPS] bits. The capture timer (QCTMR) value is latched into the capture period register (QCPRD) on every unit position event and then the capture timer is reset, a flag is set in QEPSTS[UPEVNT] to indicate that new value is latched into the QCPRD register. Software can check this status flag before reading the period register for low speed measurement and clear the flag by writing 1.

Time measurement (ΔT) between unit position events will be correct if the following conditions are met:

- No more than 65,535 counts have occurred between unit position events.
- No direction change between unit position events.

The capture unit sets the eQEP overflow error flag (QEPSTS[COEF]) in the event of capture timer overflow between unit position events. If a direction change occurs between the unit position events, then an error flag is set in the status register (QEPSTS[CDEF]).

Capture Timer (QCTMR) and Capture period register (QCPRD) can be configured to latch on following events.

- CPU read of QPOSCNT register
- Unit time-out event

If the QEPCTL[QCLM] bit is cleared, then the capture timer and capture period values are latched into the QCTMRLAT and QCPRDLAT registers, respectively, when the CPU reads the position counter (QPOSCNT).

If the QEPCTL[QCLM] bit is set, then the position counter, capture timer, and capture period values are latched into the QPOSLAT, QCTMRLAT and QCPRDLAT registers, respectively, on unit time out.

Figure 14-111 shows the capture unit operation along with the position counter.

NOTE: The QCAPCTL register should not be modified dynamically (such as switching CAPCLK prescaling mode from QCLK/4 to QCLK/8). The capture unit must be disabled before changing the prescaler.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 14-109. eQEP Edge Capture Unit

Figure 14-110. Unit Position Event for Low Speed Measurement (QCAPCTL[UPPS] = 0010)

N - Number of quadrature periods selected using QCAPCTL[UPPS] bits

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 14-111. eQEP Edge Capture Unit - Timing Details

Velocity Calculation Equations:

$$v(k) = \frac{x(k) - x(k-1)}{T} = \frac{\Delta X}{T} \text{ o} \quad (4)$$

where

v(k): Velocity at time instant k

x(k): Position at time instant k

x(k-1): Position at time instant k - 1

T: Fixed unit time or inverse of velocity calculation rate

ΔX: Incremental position movement in unit time

X: Fixed unit position

ΔT: Incremental time elapsed for unit position movement

t(k): Time instant "k"

t(k-1): Time instant "k - 1"

Unit time (T) and unit period (X) are configured using the QUPRD and QCAPCTL[UPPS] registers. Incremental position output and incremental time output is available in the QPOSLAT and QCPRDLAT registers.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Parameter	Relevant Register to Configure or Read the Information
T	Unit Period Register (QUPRD)
ΔX	Incremental Position = QPOSLAT(k) - QPOSLAT(K - 1)
X	Fixed unit position defined by sensor resolution and ZCAPCTL[UPPS] bits
ΔT	Capture Period Latch (QCPRDLAT)

14.4.2.6 eQEP Watchdog

The eQEP peripheral contains a 16-bit watchdog timer that monitors the quadrature-clock to indicate proper operation of the motion-control system. The eQEP watchdog timer is clocked from SYSCLKOUT/64 and the quadrate clock event (pulse) resets the watchdog timer. If no quadrature-clock event is detected until a period match (QWDPRD = QWDTMR), then the watchdog timer will time out and the watchdog interrupt flag will be set (QFLG[WTO]). The time-out value is programmable through the watchdog period register (QWDPRD).

Figure 14-112. eQEP Watchdog Timer

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.4.2.7 Unit Timer Base

The eQEP peripheral includes a 32-bit timer (QUTMR) that is clocked by SYSCLKOUT to generate periodic interrupts for velocity calculations. The unit time out interrupt is set (QFLG[UTO]) when the unit timer (QUTMR) matches the unit period register (QUPRD).

The eQEP peripheral can be configured to latch the position counter, capture timer, and capture period values on a unit time out event so that latched values are used for velocity calculation as described in Section Section 14.4.2.5.

Figure 14-113. eQEP Unit Time Base

14.4.2.8 eQEP Interrupt Structure

Figure 14-114 shows how the interrupt mechanism works in the EQEP module.

Figure 14-114. EQEP Interrupt Generation

Eleven interrupt events (PCE, PHE, QDC, WTO, PCU, PCO, PCR, PCM, SEL, IEL, and UTO) can be generated. The interrupt control register (QEINT) is used to enable/disable individual interrupt event sources. The interrupt flag register (QFLG) indicates if any interrupt event has been latched and contains the global interrupt flag bit (INT). An interrupt pulse is generated only to the interrupt controller if any of the interrupt events is enabled, the flag bit is 1 and the INT flag bit is 0. The interrupt service routine will need to clear the global interrupt flag bit and the serviced event, via the interrupt clear register (QCLR), before any other interrupt pulses are generated. You can force an interrupt event by way of the interrupt force register (QFRC), which is useful for test purposes.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.4.3 eQEP Registers

Table 14-86 lists the registers with their memory locations, sizes, and reset values.

Table 14-86. eQEP Registers

Offset	Acronym	Register Description	Size(×16)/ #shadow	Section
0h	QPOSCNT	eQEP Position Counter Register	2/0	Section 14.4.3.1
4h	QPOSINIT	eQEP Position Counter Initialization Register	2/0	Section 14.4.3.2
8h	QPOSMAX	eQEP Maximum Position Count Register	2/0	Section 14.4.3.3
Ch	QPOSCMP	eQEP Position-Compare Register	2/1	Section 14.4.3.4
10h	QPOSILAT	eQEP Index Position Latch Register	2/0	Section 14.4.3.5
14h	QPOSSLAT	eQEP Strobe Position Latch Register	2/0	Section 14.4.3.6
18h	QPOSLAT	eQEP Position Counter Latch Register	2/0	Section 14.4.3.7
1Ch	QUTMR	eQEP Unit Timer Register	2/0	Section 14.4.3.8
20h	QUPRD	eQEP Unit Period Register	2/0	Section 14.4.3.9
24h	QWDTMR	eQEP Watchdog Timer Register	1/0	Section 14.4.3.10
26h	QWDPRD	eQEP Watchdog Period Register	1/0	Section 14.4.3.11
28h	QDECCTL	eQEP Decoder Control Register	1/0	Section 14.4.3.12
2Ah	QEPCTL	eQEP Control Register	1/0	Section 14.4.3.13
2Ch	QCAPCTL	eQEP Capture Control Register	1/0	Section 14.4.3.14
2Eh	QPOSCTL	eQEP Position-Compare Control Register	1/0	Section 14.4.3.15
30h	QEINT	eQEP Interrupt Enable Register	1/0	Section 14.4.3.16
32h	QFLG	eQEP Interrupt Flag Register	1/0	Section 14.4.3.17
34h	QCLR	eQEP Interrupt Clear Register	1/0	Section 14.4.3.18
36h	QFRC	eQEP Interrupt Force Register	1/0	Section 14.4.3.19
38h	QEPSTS	eQEP Status Register	1/0	Section 14.4.3.20
3Ah	QCTMR	eQEP Capture Timer Register	1/0	Section 14.4.3.21
3Ch	QCPRD	eQEP Capture Period Register	1/0	Section 14.4.3.22
3Eh	QCTMRLAT	eQEP Capture Timer Latch Register	1/0	Section 14.4.3.23
40h	QCPRDLAT	eQEP Capture Period Latch Register	1/0	Section 14.4.3.24
5Ch	REVID	eQEP Revision ID Register	2/0	Section 14.4.3.25

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.4.3.1 eQEP Position Counter Register (QPOSCNT)

Figure 14-115. eQEP Position Counter Register (QPOSCNT)

31	0
QPOSCNT	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 14-87. eQEP Position Counter Register (QPOSCNT) Field Descriptions

Bits	Name	Value	Description
31-0	QPOSCNT	0-FFFF FFFFh	This 32-bit position counter register counts up/down on every eQEP pulse based on direction input. This counter acts as a position integrator whose count value is proportional to position from a give reference point.

14.4.3.2 eQEP Position Counter Initialization Register (QPOSINIT)

Figure 14-116. eQEP Position Counter Initialization Register (QPOSINIT)

31	0
QPOSINIT	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 14-88. eQEP Position Counter Initialization Register (QPOSINIT) Field Descriptions

Bits	Name	Value	Description
31-0	QPOSINIT	0-FFFF FFFFh	This register contains the position value that is used to initialize the position counter based on external strobe or index event. The position counter can be initialized through software.

14.4.3.3 eQEP Maximum Position Count Register (QPOSMAX)

Figure 14-117. eQEP Maximum Position Count Register (QPOSMAX)

31	0
QPOSMAX	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 14-89. eQEP Maximum Position Count Register (QPOSMAX) Field Descriptions

Bits	Name	Value	Description
31-0	QPOSMAX	0-FFFF FFFFh	This register contains the maximum position counter value.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.4.3.4 eQEP Position-Compare Register (QPOSCMP)

Figure 14-118. eQEP Position-Compare Register (QPOSCMP)

31	0
QPOSCMP	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 14-90. eQEP Position-Compare Register (QPOSCMP) Field Descriptions

Bits	Name	Value	Description
31-0	QPOSCMP	0-FFFF FFFFh	The position-compare value in this register is compared with the position counter (QPOSCNT) to generate sync output and/or interrupt on compare match.

14.4.3.5 eQEP Index Position Latch Register (QPOSILAT)

Figure 14-119. eQEP Index Position Latch Register (QPOSILAT)

31	0
QPOSILAT	
R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 14-91. eQEP Index Position Latch Register (QPOSILAT) Field Descriptions

Bits	Name	Value	Description
31-0	QPOSILAT	0-FFFF FFFFh	The position-counter value is latched into this register on an index event as defined by the QEPCTL[IEL] bits.

14.4.3.6 eQEP Strobe Position Latch Register (QPOSSLAT)

Figure 14-120. eQEP Strobe Position Latch Register (QPOSSLAT)

31	0
QPOSSLAT	
R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 14-92. eQEP Strobe Position Latch Register (QPOSSLAT) Field Descriptions

Bits	Name	Value	Description
31-0	QPOSSLAT	0-FFFF FFFFh	The position-counter value is latched into this register on strobe event as defined by the QEPCTL[SEL] bits.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.4.3.7 eQEP Position Counter Latch Register (QPOSLAT)

Figure 14-121. eQEP Position Counter Latch Register (QPOSLAT)

31	0
QPOSLAT	
R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 14-93. eQEP Position Counter Latch Register (QPOSLAT) Field Descriptions

Bits	Name	Value	Description
31-0	QPOSLAT	0-FFFF FFFFh	The position-counter value is latched into this register on unit time out event.

14.4.3.8 eQEP Unit Timer Register (QUTMR)

Figure 14-122. eQEP Unit Timer Register (QUTMR)

31	0
QUTMR	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 14-94. eQEP Unit Timer Register (QUTMR) Field Descriptions

Bits	Name	Value	Description
31-0	QUTMR	0-FFFF FFFFh	This register acts as time base for unit time event generation. When this timer value matches with unit time period value, unit time event is generated.

14.4.3.9 eQEP Unit Period Register (QUPRD)

Figure 14-123. eQEP Unit Period Register (QUPRD)

31	0
QUPRD	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 14-95. eQEP Unit Period Register (QUPRD) Field Descriptions

Bits	Name	Value	Description
31-0	QUPRD	0-FFFF FFFFh	This register contains the period count for unit timer to generate periodic unit time events to latch the eQEP position information at periodic interval and optionally to generate interrupt.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.4.3.10 eQEP Watchdog Timer Register (QWDTMR)

Figure 14-124. eQEP Watchdog Timer Register (QWDTMR)

15	0
QWDTMR	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 14-96. eQEP Watchdog Timer Register (QWDTMR) Field Descriptions

Bits	Name	Value	Description
15-0	QWDTMR	0-FFFF FFFFh	This register acts as time base for watch dog to detect motor stalls. When this timer value matches with watch dog period value, watch dog timeout interrupt is generated. This register is reset upon edge transition in quadrature-clock indicating the motion.

14.4.3.11 eQEP Watchdog Period Register (QWDPRD)

Figure 14-125. eQEP Watchdog Period Register (QWDPRD)

15	0
QWDPRD	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 14-97. eQEP Watchdog Period Register (QWDPRD) Field Description

Bits	Name	Value	Description
15-0	QWDPRD	0-FFFFh	This register contains the time-out count for the eQEP peripheral watch dog timer. When the watchdog timer value matches the watchdog period value, a watchdog timeout interrupt is generated.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.4.3.12 QEP Decoder Control Register (QDECCTL)

Figure 14-126. QEP Decoder Control Register (QDECCTL)

15	14	13	12	11	10	9	8
QSRC		SOEN	SPSEL	XCR	SWAP	IGATE	QAP
R/W-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

7	6	5	4	0
QBP	QIP	QSP	Reserved	
R/W-0	R/W-0	R/W-0	R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 14-98. eQEP Decoder Control Register (QDECCTL) Field Descriptions

Bits	Name	Value	Description
15-14	QSRC	0-3h	Position-counter source selection
		0	Quadrature count mode (QCLK = iCLK, QDIR = iDIR)
		1h	Direction-count mode (QCLK = xCLK, QDIR = xDIR)
		2h	UP count mode for frequency measurement (QCLK = xCLK, QDIR = 1)
		3h	DOWN count mode for frequency measurement (QCLK = xCLK, QDIR = 0)
13	SOEN		Sync output-enable
		0	Disable position-compare sync output
		1	Enable position-compare sync output
12	SPSEL		Sync output pin selection
		0	Index pin is used for sync output
		1	Strobe pin is used for sync output
11	XCR		External clock rate
		0	2× resolution: Count the rising/falling edge
		1	1× resolution: Count the rising edge only
10	SWAP		Swap quadrature clock inputs. This swaps the input to the quadrature decoder, reversing the counting direction.
		0	Quadrature-clock inputs are not swapped
		1	Quadrature-clock inputs are swapped
9	IGATE		Index pulse gating option
		0	Disable gating of Index pulse
		1	Gate the index pin with strobe
8	QAP		QEPA input polarity
		0	No effect
		1	Negates QEPA input
7	QBP		QEPB input polarity
		0	No effect
		1	Negates QEPB input
6	QIP		QEPI input polarity
		0	No effect
		1	Negates QEPI input
5	QSP		QEPS input polarity
		0	No effect
		1	Negates QEPS input
4-0	Reserved	0	Always write as 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.4.3.13 eQEP Control Register (QEPCTL)

Figure 14-127. eQEP Control Register (QEPCTL)

15	14	13	12	11	10	9	8
FREE, SOFT		PCRM		SEI		IEI	
R/W-0		R/W-0		R/W-0		R/W-0	

7	6	5	4	3	2	1	0
SWI	SEL	IEL		PHEN	QCLM	UTE	WDE
R/W-0	R/W-0	R/W-0		R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 14-99. eQEP Control Register (QEPCTL) Field Descriptions

Bits	Name	Value	Description
15-14	FREE, SOFT	0-3h	Emulation Control Bits
			QPOSCNT behavior:
		0	Position counter stops immediately on emulation suspend.
		1h	Position counter continues to count until the rollover.
		2h-3h	Position counter is unaffected by emulation suspend.
			QWDTMR behavior:
		0	Watchdog counter stops immediately.
		1	Watchdog counter counts until WD period match roll over.
		2h-3h	Watchdog counter is unaffected by emulation suspend.
			QUTMR behavior:
		0	Unit timer stops immediately.
		1h	Unit timer counts until period rollover.
		2h-3h	Unit timer is unaffected by emulation suspend.
			QCTMR behavior:
		0	Capture Timer stops immediately.
		1h	Capture Timer counts until next unit period event.
		2h-3h	Capture Timer is unaffected by emulation suspend.
13-12	PCRM	0-3h	Position counter reset mode
		0	Position counter reset on an index event
		1h	Position counter reset on the maximum position
		2h	Position counter reset on the first index event
		3h	Position counter reset on a unit time event
11-10	SEI	0-3h	Strobe event initialization of position counter
		0	Does nothing (action disabled)
		1h	Does nothing (action disabled)
		2h	Initializes the position counter on rising edge of the QEPS signal
		3h	**Clockwise Direction:** Initializes the position counter on the rising edge of QEPS strobe
			Counter Clockwise Direction: Initializes the position counter on the falling edge of QEPS strobe
9-8	IEI	0-3h	Index event initialization of position counter
		0	Do nothing (action disabled)
		1h	Do nothing (action disabled)
		2h	Initializes the position counter on the rising edge of the QEPI signal (QPOSCNT = QPOSINIT)
		3h	Initializes the position counter on the falling edge of QEPI signal (QPOSCNT = QPOSINIT)
7	SWI		Software initialization of position counter
		0	Do nothing (action disabled)
		1	Initialize position counter, this bit is cleared automatically

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-99. eQEP Control Register (QEPCTL) Field Descriptions (continued)

Bits	Name	Value	Description
6	SEL		Strobe event latch of position counter
		0	The position counter is latched on the rising edge of QEPS strobe (QPOSSLAT = POSCCNT). Latching on the falling edge can be done by inverting the strobe input using the QSP bit in the QDECCTL register.
		1	**Clockwise Direction:** Position counter is latched on rising edge of QEPS strobe
			Counter Clockwise Direction: Position counter is latched on falling edge of QEPS strobe
5-4	IEL	0-3h	Index event latch of position counter (software index marker)
		0	Reserved
		1h	Latches position counter on rising edge of the index signal
		2h	Latches position counter on falling edge of the index signal
		3h	Software index marker. Latches the position counter and quadrature direction flag on index event marker. The position counter is latched to the QPOSILAT register and the direction flag is latched in the QEPSTS[QDLF] bit. This mode is useful for software index marking.
3	PHEN		Quadrature position counter enable/software reset
		0	Reset the eQEP peripheral internal operating flags/read-only registers. Control/configuration registers are not disturbed by a software reset.
		1	eQEP position counter is enabled
2	QCLM		eQEP capture latch mode
		0	Latch on position counter read by CPU. Capture timer and capture period values are latched into QCTMRLAT and QCPRDLAT registers when CPU reads the QPOSCNT register.
		1	Latch on unit time out. Position counter, capture timer and capture period values are latched into QPOSLAT, QCTMRLAT and QCPRDLAT registers on unit time out.
1	UTE		eQEP unit timer enable
		0	Disable eQEP unit timer
		1	Enable unit timer
0	WDE		eQEP watchdog enable
		0	Disable the eQEP watchdog timer
		1	Enable the eQEP watchdog timer

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.4.3.14 eQEP Capture Control Register (QCAPCTL)

Figure 14-128. eQEP Capture Control Register (QCAPCTL)

15	14 … 8
CEN	Reserved
R/W-0	R-0

7	6 … 4	3 … 0
Reserved	CCPS	UPPS
R-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 14-100. eQEP Capture Control Register (QCAPCTL) Field Descriptions

Bits	Name	Value	Description
15	CEN		Enable eQEP capture
		0	eQEP capture unit is disabled
		1	eQEP capture unit is enabled
14-7	Reserved	0	Always write as 0
6-4	CCPS	0-7h	eQEP capture timer clock prescaler
		0	CAPCLK = SYSCLKOUT/1
		1h	CAPCLK = SYSCLKOUT/2
		2h	CAPCLK = SYSCLKOUT/4
		3h	CAPCLK = SYSCLKOUT/8
		4h	CAPCLK = SYSCLKOUT/16
		5h	CAPCLK = SYSCLKOUT/32
		6h	CAPCLK = SYSCLKOUT/64
		7h	CAPCLK = SYSCLKOUT/128
3-0	UPPS	0-Fh	Unit position event prescaler
		0	UPEVNT = QCLK/1
		1h	UPEVNT = QCLK/2
		2h	UPEVNT = QCLK/4
		3h	UPEVNT = QCLK/8
		4h	UPEVNT = QCLK/16
		5h	UPEVNT = QCLK/32
		6h	UPEVNT = QCLK/64
		7h	UPEVNT = QCLK/128
		8h	UPEVNT = QCLK/256
		9h	UPEVNT = QCLK/512
		Ah	UPEVNT = QCLK/1024
		Bh	UPEVNT = QCLK/2048
		Ch-Fh	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.4.3.15 eQEP Position-Compare Control Register (QPOSCTL)

Figure 14-129. eQEP Position-Compare Control Register (QPOSCTL)

15	14	13	12	11	8
PCSHDW	PCLOAD	PCPOL	PCE	PCSPW	
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	

7	0
PCSPW	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 14-101. eQEP Position-Compare Control Register (QPOSCTL) Field Descriptions

Bit	Name	Value	Description
15	PCSHDW		Position-compare shadow enable
		0	Shadow disabled, load Immediate
		1	Shadow enabled
14	PCLOAD		Position-compare shadow load mode
		0	Load on QPOSCNT = 0
		1	Load when QPOSCNT = QPOSCMP
13	PCPOL		Polarity of sync output
		0	Active HIGH pulse output
		1	Active LOW pulse output
12	PCE		Position-compare enable/disable
		0	Disable position compare unit
		1	Enable position compare unit
11-0	PCSPW	0-FFFh	Select-position-compare sync output pulse width
		0	1 × 4 × SYSCLKOUT cycles
		1h	2 × 4 × SYSCLKOUT cycles
		2h-FFFh	3 × 4 × SYSCLKOUT cycles to 4096 × 4 × SYSCLKOUT cycles

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.4.3.16 eQEP Interrupt Enable Register (QEINT)

Figure 14-130. eQEP Interrupt Enable Register (QEINT)

15			12	11	10	9	8
Reserved				UTO	IEL	SEL	PCM
R-0				R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
PCR	PCO	PCU	WTO	QDC	PHE	PCE	Reserved
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 14-102. eQEP Interrupt Enable Register (QEINT) Field Descriptions

Bits	Name	Value	Description
15-12	Reserved	0	Always write as 0
11	UTO		Unit time out interrupt enable
		0	Interrupt is disabled
		1	Interrupt is enabled
10	IEL		Index event latch interrupt enable
		0	Interrupt is disabled
		1	Interrupt is enabled
9	SEL		Strobe event latch interrupt enable
		0	Interrupt is disabled
		1	Interrupt is enabled
8	PCM		Position-compare match interrupt enable
		0	Interrupt is disabled
		1	Interrupt is enabled
7	PCR		Position-compare ready interrupt enable
		0	Interrupt is disabled
		1	Interrupt is enabled
6	PCO		Position counter overflow interrupt enable
		0	Interrupt is disabled
		1	Interrupt is enabled
5	PCU		Position counter underflow interrupt enable
		0	Interrupt is disabled
		1	Interrupt is enabled
4	WTO		Watchdog time out interrupt enable
		0	Interrupt is disabled
		1	Interrupt is enabled
3	QDC		Quadrature direction change interrupt enable
		0	Interrupt is disabled
		1	Interrupt is enabled
2	PHE		Quadrature phase error interrupt enable
		0	Interrupt is disabled
		1	Interrupt is enabled
1	PCE		Position counter error interrupt enable
		0	Interrupt is disabled
		1	Interrupt is enabled
0	Reserved	0	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.4.3.17 eQEP Interrupt Flag Register (QFLG)

Figure 14-131. eQEP Interrupt Flag Register (QFLG)

15			12	11	10	9	8
Reserved				UTO	IEL	SEL	PCM
R-0				R-0	R-0	R-0	R-0
7	6	5	4	3	2	1	0
PCR	PCO	PCU	WTO	QDC	PHE	PCE	INT
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 14-103. eQEP Interrupt Flag Register (QFLG) Field Descriptions

Bits	Name	Value	Description
15-12	Reserved	0	Always write as 0
11	UTO		Unit time out interrupt flag
		0	No interrupt generated
		1	Set by eQEP unit timer period match
10	IEL		Index event latch interrupt flag
		0	No interrupt generated
		1	This bit is set after latching the QPOSCNT to QPOSILAT
9	SEL		Strobe event latch interrupt flag
		0	No interrupt generated
		1	This bit is set after latching the QPOSCNT to QPOSSLAT
8	PCM		eQEP compare match event interrupt flag
		0	No interrupt generated
		1	This bit is set on position-compare match
7	PCR		Position-compare ready interrupt flag
		0	No interrupt generated
		1	This bit is set after transferring the shadow register value to the active position compare register.
6	PCO		Position counter overflow interrupt flag
		0	No interrupt generated
		1	This bit is set on position counter overflow.
5	PCU		Position counter underflow interrupt flag
		0	No interrupt generated
		1	This bit is set on position counter underflow.
4	WTO		Watchdog timeout interrupt flag
		0	No interrupt generated
		1	Set by watch dog timeout
3	QDC		Quadrature direction change interrupt flag
		0	No interrupt generated
		1	This bit is set during change of direction
2	PHE		Quadrature phase error interrupt flag
		0	No interrupt generated
		1	Set on simultaneous transition of QEPA and QEPB
1	PCE		Position counter error interrupt flag
		0	No interrupt generated
		1	Position counter error

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-103. eQEP Interrupt Flag Register (QFLG) Field Descriptions (continued)

Bits	Name	Value	Description
0	INT		Global interrupt status flag
		0	No interrupt generated
		1	Interrupt was generated

14.4.3.18 eQEP Interrupt Clear Register (QCLR)

Figure 14-132. eQEP Interrupt Clear Register (QCLR)

15			12	11	10	9	8
Reserved				UTO	IEL	SEL	PCM
R-0				R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
PCR	PCO	PCU	WTO	QDC	PHE	PCE	INT
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 14-104. eQEP Interrupt Clear Register (QCLR) Field Descriptions

Bit	Field	Value	Description
15-12	Reserved	0	Always write as 0s
11	UTO		Clear unit time out interrupt flag
		0	No effect
		1	Clears the interrupt flag
10	IEL		Clear index event latch interrupt flag
		0	No effect
		1	Clears the interrupt flag
9	SEL		Clear strobe event latch interrupt flag
		0	No effect
		1	Clears the interrupt flag
8	PCM		Clear eQEP compare match event interrupt flag
		0	No effect
		1	Clears the interrupt flag
7	PCR		Clear position-compare ready interrupt flag
		0	No effect
		1	Clears the interrupt flag
6	PCO		Clear position counter overflow interrupt flag
		0	No effect
		1	Clears the interrupt flag
5	PCU		Clear position counter underflow interrupt flag
		0	No effect
		1	Clears the interrupt flag
4	WTO		Clear watchdog timeout interrupt flag
		0	No effect
		1	Clears the interrupt flag
3	QDC		Clear quadrature direction change interrupt flag
		0	No effect
		1	Clears the interrupt flag

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 14-104. eQEP Interrupt Clear Register (QCLR) Field Descriptions (continued)

Bit	Field	Value	Description
2	PHE		Clear quadrature phase error interrupt flag
		0	No effect
		1	Clears the interrupt flag
1	PCE		Clear position counter error interrupt flag
		0	No effect
		1	Clears the interrupt flag
0	INT		Global interrupt clear flag
		0	No effect
		1	Clears the interrupt flag and enables further interrupts to be generated if an event flags is set to 1.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.4.3.19 eQEP Interrupt Force Register (QFRC)

Figure 14-133. eQEP Interrupt Force Register (QFRC)

15			12	11	10	9	8
Reserved				UTO	IEL	SEL	PCM
R-0				R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
PCR	PCO	PCU	WTO	QDC	PHE	PCE	Reserved
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 14-105. eQEP Interrupt Force Register (QFRC) Field Descriptions

Bit	Field	Value	Description
15-12	Reserved	0	Always write as 0s
11	UTO		Force unit time out interrupt
		0	No effect
		1	Force the interrupt
10	IEL		Force index event latch interrupt
		0	No effect
		1	Force the interrupt
9	SEL		Force strobe event latch interrupt
		0	No effect
		1	Force the interrupt
8	PCM		Force position-compare match interrupt
		0	No effect
		1	Force the interrupt
7	PCR		Force position-compare ready interrupt
		0	No effect
		1	Force the interrupt
6	PCO		Force position counter overflow interrupt
		0	No effect
		1	Force the interrupt
5	PCU		Force position counter underflow interrupt
		0	No effect
		1	Force the interrupt
4	WTO		Force watchdog time out interrupt
		0	No effect
		1	Force the interrupt
3	QDC		Force quadrature direction change interrupt
		0	No effect
		1	Force the interrupt
2	PHE		Force quadrature phase error interrupt
		0	No effect
		1	Force the interrupt
1	PCE		Force position counter error interrupt
		0	No effect
		1	Force the interrupt
0	Reserved	0	Always write as 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.4.3.20 eQEP Status Register (QEPSTS)

Figure 14-134. eQEP Status Register (QEPSTS)

15							8
Reserved							
R-0							
7	6	5	4	3	2	1	0
UPEVNT	FIDF	QDF	QDLF	COEF	CDEF	FIMF	PCEF
R-0	R-0	R-0	R-0	R/W-1	R/W-1	R/W-1	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 14-106. eQEP Status Register (QEPSTS) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Always write as 0
7	UPEVNT		Unit position event flag
		0	No unit position event detected
		1	Unit position event detected. Write 1 to clear.
6	FDF		Direction on the first index marker. Status of the direction is latched on the first index event marker.
		0	Counter-clockwise rotation (or reverse movement) on the first index event
		1	Clockwise rotation (or forward movement) on the first index event
5	QDF		Quadrature direction flag
		0	Counter-clockwise rotation (or reverse movement)
		1	Clockwise rotation (or forward movement)
4	QDLF		eQEP direction latch flag. Status of direction is latched on every index event marker.
		0	Counter-clockwise rotation (or reverse movement) on index event marker
		1	Clockwise rotation (or forward movement) on index event marker
3	COEF		Capture overflow error flag
		0	Sticky bit, cleared by writing 1
		1	Overflow occurred in eQEP Capture timer (QEPCTMR)
2	CDEF		Capture direction error flag
		0	Sticky bit, cleared by writing 1
		1	Direction change occurred between the capture position event.
1	FIMF		First index marker flag
		0	Sticky bit, cleared by writing 1
		1	Set by first occurrence of index pulse
0	PCEF		Position counter error flag. This bit is not sticky and it is updated for every index event.
		0	No error occurred during the last index transition.
		1	Position counter error

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.4.3.21 eQEP Capture Timer Register (QCTMR)

Figure 14-135. eQEP Capture Timer Register (QCTMR)

15	0
QCTMR	
R/W	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 14-107. eQEP Capture Time Register (QCTMR) Field Descriptions

Bits	Name	Value	Description
15-0	QCTMR	0-FFFFh	This register provides time base for edge capture unit.

14.4.3.22 eQEP Capture Period Register (QCPRD)

Figure 14-136. eQEP Capture Period Register (QCPRD)

15	0
QCPRD	
R/W	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 14-108. eQEP Capture Period Register (QCPRD) Field Descriptions

Bits	Name	Value	Description
15-0	QCPRD	0-FFFFh	This register holds the period count value between the last successive eQEP position events

14.4.3.23 eQEP Capture Timer Latch Register (QCTMRLAT)

Figure 14-137. eQEP Capture Timer Latch Register (QCTMRLAT)

15	0
QCTMRLAT	
R	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 14-109. eQEP Capture Timer Latch Register (QCTMRLAT) Field Descriptions

Bits	Name	Value	Description
15-0	QCTMRLAT	0-FFFFh	The eQEP capture timer value can be latched into this register on two events viz., unit timeout event, reading the eQEP position counter.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

14.4.3.24 eQEP Capture Period Latch Register (QCPRDLAT)

Figure 14-138. eQEP Capture Period Latch Register (QCPRDLAT)

15	0
QCPRDLAT	
R/W	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 14-110. eQEP Capture Period Latch Register (QCPRDLAT) Field Descriptions

Bits	Name	Value	Description
15-0	QCPRDLAT	0-FFFFh	eQEP capture period value can be latched into this register on two events viz., unit timeout event, reading the eQEP position counter.

14.4.3.25 eQEP Revision ID Register (REVID)

Figure 14-139. eQEP Revision ID Register (REVID)

31	0
REV	
R-44D3 1103h	

LEGEND: R = Read only; -*n* = value after reset

Table 14-111. eQEP Revision ID Register (REVID) Field Descriptions

Bits	Name	Value	Description
31-0	REV	44D3 1103h	eQEP revision ID

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Chapter 15

SPRUH73–July 2011

Universal Serial Bus (USB)

This chapter describes the USB of the device.

Topic **Page**

15.1 Introduction **2048**
15.2 Integration **2051**
15.3 Functional Description **2053**
15.4 Supported Use Cases **2115**
15.5 USB Registers **2116**

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.1 Introduction

The USB controller provides a low-cost connectivity solution for numerous consumer portable devices by providing a mechanism for data transfer between USB devices with a line/bus speed up to 480 Mbps. The device USB subsystem has two independent USB 2.0 Modules built around two OTG controllers. The OTG supplement feature, the support for a dynamic role change, is also supported. Each port has the support for a dual-role feature allowing for additional versatility enabling operation capability as a host or peripheral. Both ports have identical capabilities and operate independent of each other.

Each USB controller is built around the Mentor USB OTG controller (musbmhdrc) and TI GS70 PHY. Each USB controller has user configurable 32K Bytes of Endpoint FIFO and has the support for 15 'Transmit' endpoints and 15 'Receive' endpoints in addition to Endpoint 0. The USB subsystem makes use of the CPPI 4.1 DMA for accelerating data movement via a dedicated DMA hardware.

The two USB modules share the CPPI DMA controller and accompanying Queue Manager, Interrupt Pacer, Power Management module, and PHY/UTMI clock.

Within the descriptions of the USB subsystem that would follow, the term USB controller or USB PHY is used to mean/refer to any of the two USB controllers or PHYs existing within the USB subsystem. The term USB module is used to mean/refer to any of the two USB modules. USB0 is used to refer to one of the USB modules and USB1 is used to refer to the other USB module.

15.1.1 Acronyms, Abbreviations, and Definitions

AHB	Advanced High-performance Bus
CBA	Common Bus Architecture
CDC	Change Data Capture
CPPI	Communications Port Programming Interface
CPU	Central Processing Unit
DFT	Design for Test
DMA	Direct Memory Access
DV	Design Verification
EOI	End of Interrupt
EOP	End of Packet
FIFO	First-In First-Out
FS	Full-Speed USB data rate
HNP	Host Negotiation
HS	High-Speed USB data rate
INTD	Interrupt Distributor
IP	Intellectual Property
ISO	Isochronous transfer type
LS	Low-Speed USB data rate
MHz	Megahertz
MOP	Middle of Packet
OCP	Open Core Protocol
OCP HD	OCP High Performance
OCP MMR	OCP Memory Mapped Registers
OTG	On-The-Go
PDR	Physical Design Requirements
PHY	Physical Layer Device
PPU	Packet Processing Unit
RAM	Random Access Memory
RNDIS	Remote Network Driver Interface Specification

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

RX	Receive
SCR	Switched Central Resource
SOC	System On a Chip
SOP	Start of Packet
SRP	Session Resume
TX	Transmit
USB	Universal Serial Bus
USB0	One of the two USB 2.0 Compliant USB Module
USB1	One of the two USB2.0 Compliant USB Module
USBSS	USB Subsystem (contains USB0 and USB1)
UTMI	USB 2.0 Transceiver Macrocell Interface
XDMA	Transfer DMA (DMA other than CPPI DMA used within the Controller)

15.1.1.1 USB Features

The main features of the USB subsystem are:

- Contains 2 usb20otg_f controller modules with the following features:
  - Built around the Mentor USB 2.0 OTG core (musbmhdrc)
  - Supports USB 2.0 peripheral at speeds HS (480 Mb/s) and FS (12 Mb/s)
  - Supports USB 2.0 host or OTG at speeds HS (480 Mb/s), FS (12 Mb/s), and LS (1.5 Mb/s)
  - Supports all modes of transfers (control, bulk, interrupt, and isochronous)
  - Supports high bandwidth ISO mode
  - Supports 16 Transmit (TX) and 16 Receive (RX) endpoints including endpoint 0
  - Supports USB OTG extensions for Session Resume (SRP) and Host Negotiation (HNP)
  - Includes a 32K endpoint FIFO RAM, and supports programmable FIFO sizes
  - Includes RNDIS mode for accelerating RNDIS type protocols using short packet termination over USB
  - Includes CDC Linux mode for accelerating CDC type protocols using short packet termination over USB
  - Includes an RNDIS like mode for terminating RNDIS type protocols without using short packet termination for support of MSC applications
- Includes 2 GS70 USB2.0 OTG PHYs in C014.M process
- Interfaces to the CPU via 3 OCP interfaces:
  - Master OCP HP interface for the DMA
  - Master OCP HP interface for the Queue manager
  - Slave OCP MMR interface
- Includes a CPPI 4.1 compliant DMA controller sub-module with 30 RX and 30 TX simultaneous data connections
- Includes a CPPI 4.1 DMA scheduler
- DMA supports CPPI host descriptor formats
- DMA supports stall on buffer starvation
- Supports data buffer sizes up to 4M bytes
- CPPI FIFO interface per TX/RX endpoint
- Provides a CPPI Queue Manager module with 92 queues for queuing/de-queuing packets.
- DMA pacing logic for interrupts
- Loopback MGC test using the UTMI interfaces

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.1.2 Unsupported USB OTG and PHY Features

This device supports USBOTG module host, device and OTG features.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.2 Integration

This device implements the USB2.0 OTG dual port module and PHY for interfacing to USB as a peripheral or host.

Figure 15-1 shows the integration of the USB module on this device.

Figure 15-1. USB Integration

15.2.1 USB Connectivity Attributes

The USB module itself has a very large number of interrupt outputs. For ease of integration, these outputs are all routed to a pair of interrupt aggregators. These aggregator blocks generate a single interrupt on the first occurrence of any interrupt from the USB or DMA logic of the module.

Table 15-1. USB Connectivity Attributes

Attributes	Type
Power domain	Peripheral Domain
Clock domain	PD_PER_L3S_GCLK (OCP) CLKDCOLDO (PHY)
Reset signals	DEF_DOM_RST_N USB_POR_N
Idle/Wakeup signals	Smart idle Wakeup
Interrupt request	4 interrupts usbss to MPU subsystem usb0 to MPU subsystem usb1 to MPU subsystem slv0p_Swakeup to MPU subsystem and WakeM3 2 Wakeup Events to WakeM3 usb0_wuout usb1_wuout
DMA request	None
Physical address	L3 Slow slave port

15.2.2 USB Clock and Reset Management

Each USB controller has a PHY module that generates the UTMI clock. The UTMI clock is fixed in the UTMI specification for an 8-bit interface at 60 MHz (480 Mb/s). The PHYs require a low-jitter 960-MHz source clock.

Table 15-2. USB Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
ocp_clk OCP / Functional clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_l3s_gclk From PRCM
phy0_other_refclk960m Phy reference clock	960 MHz	CLKDCOLDO	clkdcoldo_po From Per PLL
phy1_other_refclk960m Phy reference clock	960 MHz	CLKDCOLDO	clkdcoldo_po From Per PLL

15.2.3 USB Pin List

The USB external interface signals are shown in Table 15-3.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-3. USB Pin List

Pin	Type*	Description
USBx_DP USBx_DM	I/O	USBx data differential pair
USBx_DRVVBUS	O	USBx VBUS supply control
USBx_VBUS	I	USBx VBUS (input only for voltage sensing)
USBx_ID	I	USBx OTG identification
USBx_CE	O	USBx Phy charge enable

15.2.4 USB GPIO Details

The USB module supports configuration of the DP and DM pins as pass-through GPIOs. In this device, the GPIO mode is used to provide UART over USB interface functionality. Chip level logic allows connection of the UART TX/RX data signals to either DP or DM in either normal or inverted state as shown in Figure 15-2. The diagram shows the UART2 / USB0 port implementation. This logic is also replicated for the UART3 / USB1 ports.

Figure 15-2. USB GPIO Integration

15.2.5 USB Unbonded PHY Pads

The USBOTGSS includes two USB PHY modules. On packages where only 1 PHY is bonded out to pins (e.g. 13x13 package), the following procedures must be followed in order to ensure that the unbonded PHY pads do not cause issues with USBOTGSS operation:

- The USB Controller corresponding to the unbonded PHY must be placed in Host Mode by setting Bit2 of the USB Core DEVCTL register.
- The unbonded PHY must be placed in the SUSPEND state by setting Bits[1:0] of the USB Core POWER register.
- The Control Module USB_CTRLx register corresponding to the unbonded PHY must have the following bits programmed as shown:
  - CHGDET_DIS = 1
  - CM_PWRDN = 1
  - GPIOMODE = 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.3 Functional Description

The USB controller within the device supports 15 Transmit endpoints and 15 Receive endpoints in addition to Endpoint 0. (The use of these endpoints for IN and OUT transactions depends on whether the USB controller is being used as a device/peripheral or as a host. When used as a peripheral, IN transactions are processed through TX endpoints and OUT transactions are processed through Rx endpoints. When used as a host, IN transactions is processed through Rx endpoints and OUT transactions is processed through TX endpoints.)

These additional endpoints can be individually configured in software to handle either Bulk transfers (which also allows them to handle Interrupt transfers), Isochronous transfers or Control transfers. Further, the endpoints can also be allocated to different target device functions on the fly – maximizing the number of devices that can be simultaneously supported.

Each endpoint requires a chunk of memory allocated within the FIFO to be associated with it. The USB module has a single block of FIFO RAM (32 Kbytes in size) to be shared by all endpoints. The FIFO for Endpoint 0 is required to be 64 bytes deep and will buffer 1 packet. The first 64 Bytes of the FIFO RAM is reserved by hardware for Endpoint 0 usage and for this reason user software does not need (nor has the capability) to allocate FIFO for Endpoint 0. The rest of the FIFO RAM is user configurable with regard to the other endpoint FIFOs, which may be from 8 to 8192 bytes in size and can buffer either 1 or 2 packets.

Separate FIFOs may be associated with each endpoint: alternatively a TX endpoint and the Rx endpoint with the same Endpoint number can be configured to use the same FIFO, for example to reduce the size of RAM block needed, provided they can never be active at the same time.

The role (host or device) that the USB controller assumes is chosen by user firmware programming the respective USB controller MODE register defined within the USB subsystem space. Note that some USB pins have not been bonded out and the functions of these pins are controlled by user software via dedicated registers.

The user has access to the controller through the OCP slave interface via the CPU. The user can process USB transactions entirely from the CPU or can also use the DMA to perform data transfer. The CPPI DMA can be used to service Endpoints 1 to 15 not Endpoint 0. CPU access method is used to service Endpoint 0 transactions.

15.3.1 VBUS Voltage Sourcing Control

When any of the USB controllers assumes the role of a host, the USB is required to supply a 5V power source to an attached device through its VBUS line. In order to achieve this task, the USB controller requires the use of an external power logic (or charge pump) capable of sourcing 5V power. A USB_DRVVBUS is used as a control signal to enable/disable this external power logic to either source or disable power on the VBUS line. The control on the USB_DRVVBUS is automatic and is handled by the USB controller. The control should be transparent to the user so long as the proper hardware connection and software initialization are in place. The USB controller drives the USB_DRVVBUS signal high when it assumes the role of a host while the controller is in session. When assuming the role of a device, the controller drives the USB_DRVVBUS signal low disabling the external charge pump/power logic; hence, no power is driven on the VBUS line (in this case, power is expected to be sourced by the external host).

Note that both USBs are self-powered and the device does not rely on the voltage on the VBUS line sourced by an external host for controller operation when assuming the role of a device. The power on the VBUS is used to identify the presence of a Host. It is also used to power up the pull-up on the D+ line. The USB PHY would continually monitor the voltage on the VBUS and report the status to USB controller.

15.3.2 Pull-up/Pull-Down Resistors

Since the USB controllers are dual role controllers, capable of assuming a role of a host or device, the necessary required pull-up/pull-down resistors can not exist external to the device. These pull-up/pull-down resistors exist internal to the device, within the PHY to be more specific, and are enabled and disabled based on the role the controller assumes allowing dynamic hardware configuration.

When assuming the role of a host, the data lines are pulled low by the PHY enabling the internal 15KOhms resistors. When assuming the role of a device the required 1.5Kohm pull-up resistor on the D+ line is enabled automatically to signify the USB capability to the external host as a FS device (HS operation is negotiated during reset bus condition).

Copyright © 2011, Texas Instruments Incorporated

15.3.3 Role Assuming Method

For an OTG controller, the usual method followed by the controller to assume the role of a host or a device is governed by the state of the ID pin which in turn is controlled by the USB cable connector type. The DM814x device has bonded out these ID pins and allows the control to be handled directly from the connector, i.e. the USB controller would assume the role based on the cable end inserted to the mini or micro A/B connector. An alternate method where the ID pin signal is bypassed and the firmware controls the role the controller assumes exists via register access allowing additional control for systems that do not require the use of the USB ID pin. In this case the USBnMODE [Bit7=IDDIG_MUX] needs to be programmed with a value of 1 so that Register programming will be used.

Two registers, USB0 Mode Register at offset 10E8h and USB1 Mode Register at offset 18E8h, are used for a user to select the role the USB controller assumes. The user is required to program the corresponding register prior to the USB controller is in session.

15.3.4 Clock, PLL, and PHY Initialization

Prior to configuring the USB Module Registers, the DM814x USB SubSystem and PHY are required to be released from reset, enable interconnects and controllers clocks as well as configure the USB PHY with appropriate setting. Not all registers related for this task are captured within this document. Please refer to the PRCM for clocking and Control Module for PHY configuration registers definitions.

Note: The following initialization makes use of the s/w option where USB role is controlled by firmware programming USBx Mode Register.

Clock Related Configuration:

```
*0x48180B10 &= 0xFFFFFF9F; // RSTCTRL (Release USB Module from Reset)
while ((0x48180B14 & 0x00000060)>>5)!=0x3); // Wait until Modules come out Reset
*0x48180B14 |= 0x00000060; // RSTST (Clear Reset Presence)
*0x48180514 = 0x2; // L3_SLOW_CLKSTCTRL (Enable Interconnect Clock)
*0x48180558 = 0x2; // USB_CLKCTRL (Enable USB OCP Clock)
while(((*0x48180558) & 0x70000)>>16)==0x7); // Wait until USB Module is Ready
```

USB PLL Related:

```
PLL_SUBSYS_BASE  =  0x481C5000
USB_PLL_BASE  =  PLL_SUBSYS_BASE + 0x260
Output clock frequency from PLL (according to above function call) = (((OSC
frequency/(N+1))*M)/M2) = (((20MHz/(19+1))*960)/1) = 960MHz

*(0x481C5270) = 0x00010013; // USB_PLLM2NDIV
*(0x481C5274)  = 0x000003C0; // USB_PLLMN2DIV

/* Load M2, N2 dividers of ADPLL */
*(0x481C526C) = 0x00000001; // USBPLL_TENABLEDIV
   // Toggle to Complete Load
*(0x481C526C) = 0x00000000; // USBPLL_TENABLEDIV

/* Load M, N dividers of ADPLL */
*(0x481C5268) = 0x00000001; // USBPLL_TENABLE
   // Toggle to Complete Load
*(0x481C5268) = 0x00000000; // USBPLL_TENABLE

*(0x481C5264) = ((*(0x481C5264) & 0xff7fe3ff) | 0x200a080a); // USBPLL_CLKCTRL
/* Wait for phase and freq lock */
while (((*(0x481C5284)) & 0x00000600) != 0x00000600); // USBPLL_STATUS
```

USB PHY Related

```
// Clear bits 0,1,8,9,15,16,23
*(48140620) & = ~(0x00818303); // USB_CTRL0
// Set bits 6,7,10,13,14,17,18,19,20
*(48140620)  |= 0x001E64C0;    // USB_CTRL0
```

Firmware is now ready to continue programming the USB Controllers.

Copyright © 2011, Texas Instruments Incorporated

15.3.5 Indexed and Non-Indexed Register Spaces

The USB controller provides two mechanisms for accessing endpoint control and status registers; Indexed and Non-Indexed method.

Indexed Endpoint Control/Status Register Space. This register space can be thought of as a region populated with Proxy Registers. This register space is a memory-mapped region at offset 1410h to 141Fh for USB0 and 1C10h to 1C1Fh for USB1. The endpoint register space mapped at this region is selected by programming the corresponding INDEX register (@ offsets 140Eh and 1C0Eh for USB0 and USB1 respectively) of the controller.

By programming the INDEX register with the corresponding endpoint number, the control and status register corresponding to that particular endpoint is accessible from this Indexed Region. In other words, Index Register region behaves as a proxy to access a selected endpoint registers.

Non-indexed Endpoint Control/Status Register Space. These regions are dedicated endpoint registers memory-mapped residing at offsets 1500h to 16FFh for USB0 and 1D00h to 1EFFh for USB1. Registers at offset 1500h to 150Fh belong to Endpoint 0; registers at offset 1510h to 151Fh belong to Endpoint 1… and lastly registers at offset 16F0h to 16FFh belong to Endpoint 15 for USB0. Similarly registers at offset 1D00h to 1E0Fh belong to Endpoint 0; registers at offset 1D10h to 1D1Fh belong to Endpoint 1… and finally registers at offset 1DF0h to 1DFFh belong to Endpoint 15 for USB1.

This allows the user/firmware to access an endpoint register region directly from its unique space, as specified within a non-indexed region or from a proxy space by programming the corresponding Index register.

Note: The control and status registers/regions apply to both Tx and Rx of a given Endpoint. That is, Endpoint 1 Tx and Endpoint 1 Rx registers occupy the same register space.

For detailed information about the USB controller registers, see Section 15.5.2.

15.3.6 Dynamic FIFO Sizing

Each USB module supports a total of 32K bytes of FIFO RAM to dynamically allocate FIFO to all endpoints in use. Each endpoint requires a FIFO to be associated with it. There is one set of register per endpoint available for FIFO allocation, excluding endpoint 0. FIFO configuration registers are Indexed registers and cannot be accessed directly; configuration takes place via accessing proxy registers. The INDEX register at addresses 140Eh or 1C0Eh must be set to the appropriate endpoint. In other words, the Endpoint FIFO configuration registers do not have non-indexed regions.

The allocation of FIFO space to the different endpoints requires programming for each Tx and Rx endpoint with the following information:

- The start address of the FIFO within the RAM block
- The maximum size of packet to be supported
- Whether double-buffering is required

(These last two together define the amount of space that needs to be allocated to the FIFO.)

It is the responsibility of the firmware to ensure that all the Tx and Rx endpoints (other than endpoint 0) that are active in the current USB configuration have a block of FIFO RAM allocated for their use. The maximum FIFO size for endpoint 0 required is 64 bytes deep and which is large enough to buffer one packet. For this reason, the first 64 bytes of FIFO memory is reserved for Endpoint 0 usage and this resource allocation is implied to be available at all times and no FIFO allocation is required for Endpoint 0. For endpoints other than endpoint 0, the FIFO RAM interface is configurable and must have a minimum size of 8 bytes and should be capable of buffering either 1 or 2 packets. Separate FIFOs may be associated with each endpoint: alternatively a Tx endpoint and the Rx endpoint with the same Endpoint number can be configured to use the same FIFO, to reduce the size of RAM block needed, provided they can never be active at the same time.

NOTE: The option of dynamically setting FIFO sizes only applies to Endpoints 1-15. Endpoint 0 FIFO has a fixed size (64 bytes) and a fixed location which is the first 64 Bytes of FIFO (FIFO addresses 0-63).

Copyright © 2011, Texas Instruments Incorporated

15.3.7 USB Controller Host and Peripheral Modes Operation

The two USB modules can be used in a range of different environments. They can be used as either a high-speed or a full-speed USB peripheral device attached to a conventional USB host (such as a PC) in a point-to-point type of arrangement. As a host, the USB modules can also be used with another peripheral device of any speed (high-,full-, or low-speed) in a point-to-point arrangement or can be used as the host to a range of peripheral devices in a multi-point setup; one-to-many via hub. Note that the role each USB controller is assuming is independent of each other. Two options (h/w or s/w) exist for a USB role assumption/configuration.

For a USB Peripheral configuration the user has the option of using the cable end to select the role by attaching the mini or micro b side of the cable (can consider this as a h/w option) or use the optional method where the firmware is required to program the respective USB Mode Register IDDIG bit field with a value of ‘1’ prior to the USB controller goes into session (can consider this as a s/w option).

Similarly, for a USB host configuration the user has the option of using the cable end to select the role by attaching the mini or micro a side of the cable (can consider this as a h/w option) or use the optional method where the firmware is required to program the respective USB Mode Register IDDIG bit field with a value of ‘0’ value prior to the USB controller going into session (can consider this as a s/w option).

When using the s/w option, the USB ID pin state is ignored/bypassed (this is a similar configuration where an ID pin does not exist) and the USB2.0 controller role adaptation of a host or peripheral (device) is dependent upon the state of the respective USB Mode Register IDDIG field. If the IDDIG field is programmed by the user with a value of ‘1’ prior to the USB going into session, it would assume the role of a peripheral/device. However, if the IDDIG field is programmed with the value of ‘0’, the USB controller will assume the role of a host. What this means is that the USB cable end, connector, will not be able to control the role of the OTG controller and the user needs to be aware of the firmware program setting prior to performing a USB connection.

The procedure for the USB2.0 OTG controller determining its operating modes (usb controller assuming a role of a host or a peripheral) starts when the USB 2.0 controller goes into a session. The USB 2.0 controller is in session when either it senses a voltage (>= 4.4V) on the USBx_VBUSIN pin and the controller sets its DEVCTL[SESSION] bit field or when the firmware sets the DEVCTL[SESSION] bit; assuming that it will operating as a host.

When the DEVCTL[SESSION] bit is set, the controller will start sensing the state of the Iddig signal, which in turn is controlled either by the USBx_ID pin (h/w option) or by the IDDIG bit field of the MODE Register (s/w option). If the state of the iddig signal is found to be low (IDDIG is programmed with ‘0’) then the USB2.0 controller will assume the role of a host and when it finds the iddig signal to be high (IDDIG is programmed with ‘1’) then it will assume the role of a device. Note that iddig is an internal signal that could be driven to high or low via firmware from dedicated registers.

Upon the USB controller determining its role as a host, it will drive the USBx_DRVVBUS pin high to enable the external power logic so that it start sourcing the required 5V power (must be ≥ 4.4V but to account for the voltage drop on the cable it is suggested to be in the neighborhood of 4.75V). The USB2.0 controller will then wait to for the voltage of the USBx_VBUSIN to go high. Upon sampling a little time later, if it does not see the voltage on the USBX_VBUSIN pin to be within a Vbus Valid range (>= 4.4V), it will generate a Vbus error interrupt. Assuming that the voltage level of the USBX_VBUSIN is found to within the Vbus Valid range, the USB 2.0 host controller will wait for a device to connect; that is for it to see one of its data lines USB0/1_DP/DM be pulled high.

When assuming the role of a peripheral, assuming that the firmware has set the POWER[SOFTCONN] bit, and the USBx_ID pin is floating (h/w option) or USBx Mode Register IDDIG bit field is programmed with ‘1’ (s/w option) and an external host is sourcing power on the USBx_VBUSIN line then the USB2.0 controller will set the DEVCTL[SESSION] bit instructing the controller to go into session. Not that the voltage on the USBx_VBUSIN pin must be within Vbus Valid range (i.e. greater or equal to 4.4V). When the controller goes into session, it will force the USB2.0 Controller to sense the state of the iddig signal. Once it senses iddig signal to be high, it will enable its 1.5Kohm pull-up resistor to signify the external host that it is a Full-Speed device. Note that even when operating as a High-Speed peripheral; the USB controller has to first come up as Full-Speed and then later transition to High-Speed. The USB2.0 controller will then waits for a reset signal from the external host. Then after, if High-Speed option has been selected it will negotiate for a High-Speed operation and if its request has been accepted by the host, it will enable its precision 45ohm termination resistors (to stop signal reflection) on its data lines and disable the 1.5Kohm resistor.

Copyright © 2011, Texas Instruments Incorporated

15.3.8 Protocol Description(s)

This section describes the implementation of the USB protocol(s) by the USB modules.

15.3.8.1 USB Controller Peripheral Mode Operation

The USB controller assumes the role of a peripheral when the USBx_ID pin is floating or USB Mode Register[iddig=bit8] is set to 1 (provided that iddig_mux, which is bit7 of USBnMODE is also set to 1) by the user application prior to the controller goes into session. When the USB controller go into session it will assume the role of a device.

Soft connect –After a POR or USB Module soft reset, the SOFTCONN bit of POWER register (bit 6) is cleared to 0. The controller will therefore appear disconnected until the software has set the SOFTCONN bit to 1. The application software can then choose when to set the PHY into its normal mode. Systems with a lengthy initialization procedure may use this to ensure that initialization is complete and the system is ready to perform enumeration before connecting to the USB. Once the SOFTCONN bit has been set, the software can also simulate a disconnect by clearing this bit to 0.

Entry into suspend mode –When operating as a peripheral device, the controller monitors activity on the bus and when no activity has occurred for 3 ms, it goes into Suspend mode. If the Suspend interrupt has been enabled, an interrupt will be generated at this time.

At this point, the controller can then be left active (and hence able to detect when Resume signaling occurs on the USB), or the application may arrange to disable the controller by stopping its clock. However, the controller will not then be able to detect Resume signaling on the USB if clocks are not running. If such is the case, external hardware will be needed to detect Resume signaling (by monitoring the DM and DP signals), so that the clock to the controller can be restarted.

Resume Signaling –When resume signaling occurs on the bus, first the clock to the controller must be restarted if necessary. Then the controller will automatically exit Suspend mode. If the Resume interrupt is enabled, an interrupt will be generated.

Initiating a remote wakeup –If the software wants to initiate a remote wakeup while the controller is in Suspend mode, it should set the POWER[RESUME] bit to 1. The software should leave then this bit set for approximately 10 ms (minimum of 2 ms, a maximum of 15 ms) before resetting it to 0.

NOTE: No resume interrupt will be generated when the software initiates a remote wakeup.

Reset Signaling –When reset signaling or bus condition occurs on the bus, the controller performs the following actions:

- Clears FADDR register to 0
- Clears INDEX register to 0
- Flushes all endpoint FIFOs
- Clears all controller control/status registers
- Generates a reset interrupt
- Enables all interrupts at the core level.

If the HSENA bit within the POWER register (bit 5) is set, the controller also tries to negotiate for high-speed operation. Whether high-speed operation is selected is indicated by HSMODE bit of POWER register (bit 4). When the application software receives a reset interrupt, it should close any open pipes and wait for bus enumeration to begin.

15.3.8.1.1 Control Transactions:Peripheral Mode

Endpoint 0 is the main control endpoint of the core. The software is required to handle all the standard device requests that may be sent or received via endpoint 0. These are described in Universal Serial Bus Specification, Revision 2.0, Chapter 9. The protocol for these device requests involves different numbers and types of transactions per request/command. To accommodate this, the software needs to take a state machine approach to command decoding and handling.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

The Standard Device Requests received by a USB peripheral device can be divided into three categories: Zero Data Requests (in which all the information is included in the command; no additional data is required), Write Requests (in which the command will be followed by additional data), and Read Requests (in which the device is required to send data back to the host).

This section looks at the sequence of actions that the software must perform to process these different types of device request.

NOTE: The Setup packet associated with any standard device request should include an 8-byte command. Any setup packet containing a command field other than 8 bytes will be automatically rejected by the controller.

15.3.8.1.1.1 Zero Data Requests: Peripheral Mode

Zero data requests have all their information included in the 8-byte command and require no additional data to be transferred. Examples of Zero Data standard device requests are:

- SET_FEATURE
- CLEAR_FEATURE
- SET_ADDRESS
- SET_CONFIGURATION
- SET_INTERFACE

The sequence of events will begin, as with all requests, when the software receives an endpoint 0 interrupt. The RXPKTRDY bit of PERI_CSR0 (bit 0) will also have been set. The 8-byte command should then be read from the endpoint 0 FIFO, decoded and the appropriate action taken.

For example, if the command is SET_ADDRESS, the 7-bit address value contained in the command should be written to the FADDR register at the completion of the command. The PERI_CSR0 register should be written by setting the SERV_RXPKTRDY bit (bit 6) (indicating that the command has been read from the FIFO) and also setting the DATAEND bit (bit 3) (indicating that no further data is expected for this request). The interval between setting SERV_RXPKTRDY bit and DATAEND bit should be very small to avoid getting a SETUPEND error condition. It is highly recommended to set both bits at the same time.

When the host moves to the status stage of the request, a second endpoint 0 interrupt will be generated to indicate that the request has completed. No further action is required from the software. The second interrupt is just a confirmation that the request completed successfully. For SET_ADDRESS command, the address should be written to FADDR register at the completion of the command, i.e. when the status stage interrupt is received.

If the command is an unrecognized command, or for some other reason cannot be executed, then when it has been decoded, the PERI_CSR0 register should be written to set the SERV_RXPKTRDY bit (bit 6) and to set the SENDSTALL bit (bit 5). When the host moves to the status stage of the request, the controller will send a STALL packet telling the host that the request was not executed. A second endpoint 0 interrupt will be generated and the SENTSTALL bit (bit 2 of PERI_CSR0) will be set.

If the host sends more data after the DATAEND bit has been set, then the controller will send a STALL packet automatically. An endpoint 0 interrupt will be generated and the SENTSTALL bit (bit 2 of PERI_CSR0) will be set.

NOTE: DMA is not supported for endpoint 0; endpoint 0 is always serviced via CPU.

15.3.8.1.1.2 Write Requests: Peripheral Mode

Write requests involve an additional packet (or packets) of data being sent from the host after the 8-byte command. An example of a Write standard device request is: SET_DESCRIPTOR.

The sequence of events will begin, as with all requests, when the software receives an endpoint 0 interrupt. The RXPKTRDY bit of PERI_CSR0 will also have been set. The 8-byte command should then be read from the Endpoint 0 FIFO and decoded.

As with a zero data request, the PERI_CSR0 register should then be written to set the SERV_RXPKTRDY bit (bit 6) (indicating that the command has been read from the FIFO) but in this case the DATAEND bit (bit 3) should not be set (indicating that more data is expected).

Copyright © 2011, Texas Instruments Incorporated

When a second endpoint 0 interrupt is received, the PERI_CSR0 register should be read to check the endpoint status. The RXPKTRDY bit of PERI_CSR0 should be set to indicate that a data packet has been received. The COUNT0 register should then be read to determine the size of this data packet. The data packet can then be read from the endpoint 0 FIFO.

If the length of the data associated with the request (indicated by the wLength field in the command) is greater than the maximum packet size for endpoint 0, further data packets will be sent. In this case, PERI_CSR0 should be written to set the SERV_RXPKTRDY bit, but the DATAEND bit should not be set.

When all the expected data packets have been received, the PERI_CSR0 register should be written to set the SERV_RXPKTRDY bit and to set the DATAEND bit (indicating that no more data is expected).

When the host moves to the status stage of the request, another endpoint 0 interrupt will be generated to indicate that the request has completed. No further action is required from the software, the interrupt is just a confirmation that the request completed successfully.

If the command is an unrecognized command, or for some other reason cannot be executed, then when it has been decoded, the PERI_CSR0 register should be written to set the SERV_RXPKTRDY bit (bit 6) and to set the SENDSTALL bit (bit 5). When the host sends more data, the controller will send a STALL to tell the host that the request was not executed. An endpoint 0 interrupt will be generated and the SENTSTALL bit of PERI_CSR0 (bit 2) will be set.

If the host sends more data after the DATAEND has been set, then the controller will send a STALL. An endpoint 0 interrupt will be generated and the SENTSTALL bit of PERI_CSR0 (bit 2) will be set.

15.3.8.1.1.3 Read Requests: Peripheral Mode

Read requests have a packet (or packets) of data sent from the function to the host after the 8-byte command. Examples of Read Standard Device Requests are:

- GET_CONFIGURATION
- GET_INTERFACE
- GET_DESCRIPTOR
- GET_STATUS
- SYNCH_FRAME

The sequence of events will begin, as with all requests, when the software receives an endpoint 0 interrupt. The RXPKTRDY bit of PERI_CSR0 (bit 0) will also have been set. The 8-byte command should then be read from the endpoint 0 FIFO and decoded. The PERI_CSR0 register should then be written to set the SERV_RXPKTRDY bit (bit 6) (indicating that the command has read from the FIFO).

The data to be sent to the host should then be written to the endpoint 0 FIFO. If the data to be sent is greater than the maximum packet size for endpoint 0, only the maximum packet size should be written to the FIFO. The PERI_CSR0 register should then be written to set the TXPKTRDY bit (bit 1) (indicating that there is a packet in the FIFO to be sent). When the packet has been sent to the host, another endpoint 0 interrupt will be generated and the next data packet can be written to the FIFO.

When the last data packet has been written to the FIFO, the PERI_CSR0 register should be written to set the TXPKTRDY bit and to set the DATAEND bit (bit 3) (indicating that there is no more data after this packet).

When the host moves to the status stage of the request, another endpoint 0 interrupt will be generated to indicate that the request has completed. No further action is required from the software: the interrupt is just a confirmation that the request completed successfully.

If the command is an unrecognized command, or for some other reason cannot be executed, then when it has been decoded, the PERI_CSR0 register should be written to set the SERV_RXPKTRDY bit (bit 6) and to set the SENDSTALL bit (bit 5). When the host requests data, the controller will send a STALL to tell the host that the request was not executed. An endpoint 0 interrupt will be generated and the SENTSTALL bit of PERI_CSR0 (bit 2) will be set.

If the host requests more data after DATAEND (bit 3) has been set, then the controller will send a STALL. An endpoint 0 interrupt will be generated and the SENTSTALL bit of PERI_CSR0 (bit 2) will be set.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.3.8.1.1.4 Endpoint 0 States: Peripheral Mode

When the USB controller is operating as a peripheral device, the endpoint 0 control needs three modes – IDLE, TX and RX – corresponding to the different phases of the control transfer and the states endpoint 0 enters for the different phases of the transfer (described in later sections).

The default mode on power-up or reset should be IDLE. RXPKTRDY bit of PERI_CSR0 (bit 0) becoming set when endpoint 0 is in IDLE state indicates a new device request. Once the device request is unloaded from the FIFO, the controller decodes the descriptor to find whether there is a data phase and, if so, the direction of the data phase of the control transfer (in order to set the FIFO direction). See Figure 15-3.

Depending on the direction of the data phase, endpoint 0 goes into either TX state or RX state. If there is no Data phase, endpoint 0 remains in IDLE state to accept the next device request

The actions that the CPU needs to take at the different phases of the possible transfers (for example, loading the FIFO, setting TXPKTRDY) are indicated in Figure 15-4.

Figure 15-3. CPU Actions at Transfer Phases

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 15-4. Sequence of Transfer

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.3.8.1.1.4.1 Endpoint 0 Interrupt Generating Events: Peripheral Mode

An Endpoint 0 interrupt is generated when any of the below events occur and Interrupt Service handler should determine the right event and launch the proper handler

The following events will cause Endpoint 0 Interrupt to be generated while the controller is operating assuming the role of a device/peripheral:

- The controller sets the RXPKTRDY bit of PERI_CSR0 (bit 0) after a valid token has been received and data has been written to the FIFO (legal status condition).
- The controller clears the TXPKTRDY bit of PERI_CSR0 (bit 1) after the packet of data in the FIFO has been successfully transmitted to the host (legal status condition).
- The controller sets the SENTSTALL bit of PERI_CSR0 (bit 2) after a control transaction is ended due to a protocol violation (illegal status condition).
- The controller sets the SETUPEND bit of PERI_CSR0 (bit 4) because a control transfer has ended before DATAEND (bit 3 of PERI_CSR0) is set (illegal status condition).

In order to determine the cause of the interrupt, the detail below recommends the order of cause determination and execution.

Whenever the endpoint 0 service routine is entered, the software must first check to see if the current control transfer has been ended due to either a STALL condition or a premature end of control transfer. If the control transfer ends due to a STALL condition, the SENTSTALL bit would be set. If the control transfer ends due to a premature end of control transfer, the SETUPEND bit would be set. In either case, the software should abort processing the current control transfer and set the state to IDLE.

Once the software has determined that the interrupt was not generated by an illegal bus state, the next action taken depends on the endpoint state.

If endpoint 0 is in IDLE state, the only valid reason an interrupt can be generated is as a result of the controller receiving data from the bus. The service routine must check for this by testing the RXPKTRDY bit of PERI_CSR0 (bit 0). If this bit is set, then the controller has received a SETUP packet. This must be unloaded from the FIFO and decoded to determine the action the controller must take. Depending on the command contained within the SETUP packet, endpoint 0 will enter one of three states:

- If the command is a single packet transaction (SET_ADDRESS, SET_INTERFACE etc.) without any data phase, the endpoint will remain in IDLE state.
- If the command has an OUT data phase (SET_DESCRIPTOR etc.), the endpoint will enter RX state.
- If the command has an IN data phase (GET_DESCRIPTOR etc.), the endpoint will enter TX state.
- If the endpoint 0 is in TX state, the interrupt indicates that the core has received an IN token and data from the FIFO has been sent. The software must respond to this either by placing more data in the FIFO if the host is still expecting more data or by setting the DATAEND bit to indicate that the data phase is complete. Once the data phase of the transaction has been completed, endpoint 0 should be returned to IDLE state to await the next control transaction (status stage).
- If the endpoint is in RX state, the interrupt indicates that a data packet has been received. The software must respond by unloading the received data from the FIFO. The software must then determine whether it has received all of the expected data. If it has, the software should set the DATAEND bit and return endpoint 0 to IDLE state. If more data is expected, the firmware should set the SERV_RXPKTRDY bit of PERI_CSR0 (bit 6) to indicate that it has read the data in the FIFO and leave the endpoint in RX state.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.3.8.1.1.4.2 Idle Mode: Control Transfer of Peripheral Mode

IDLE mode is the mode the Endpoint 0 control needs to select at power-on or reset and is the mode to which the Endpoint 0 control should return when the RX and TX modes are terminated.

It is also the mode in which the SETUP phase of control transfer is handled (as outlined in Figure 15-5).

Figure 15-5. Flow Chart of Setup Stage of a Control Transfer in Peripheral Mode

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.3.8.1.1.4.3 TX Mode: Control Transfer of Peripheral Mode

When the endpoint is in TX state, all arriving IN tokens need to be treated as part of a data phase until the required amount of data has been sent to the host. If either a SETUP or an OUT token is received while the endpoint is in the TX state, this will cause a SETUPEND condition to occur as the core expects only IN tokens.

Three events can cause TX mode to be terminated before the expected amount of data has been sent as shown in Figure 15-6:

1. The host sends an invalid token causing a SETUPEND condition (bit 4 of PERI_CSR0 is set).
2. The firmware sends a packet containing less than the maximum packet size for Endpoint 0.
3. The firmware sends an empty data packet.

Until the transaction is terminated, the firmware simply needs to load the FIFO when it receives an interrupt which indicates that a packet has been sent from the FIFO. An interrupt is generated when TXPKTRDY is cleared.

When the firmware forces the termination of a transfer (by sending a short or empty data packet), it should set the DATAEND bit of PERI_CSR0 (bit 3) to indicate to the core that the data phase is complete and that the core should next receive an acknowledge packet.

Figure 15-6. Flow Chart of Transmit Data Stage of a Control Transfer in Peripheral Mode

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.3.8.1.1.4.4 Rx Mode: Control Transfer of Peripheral Mode

In RX mode, all arriving data should be treated as part of a data phase until the expected amount of data has been received. If either a SETUP or an IN token is received while the endpoint is in RX state, a SETUPEND condition will occur as the controller expects only OUT tokens.

Three events can cause RX mode to be terminated before the expected amount of data has been received as shown in Figure 15-7:

1. The host sends an invalid token causing a SETUPEND condition (setting bit 4 of PERI_CSR0).
2. The host sends a packet which contains less than the maximum packet size for endpoint 0.
3. The host sends an empty data packet.

Until the transaction is terminated, the software unloads the FIFO when it receives an interrupt that indicates new data has arrived (setting RXPKTRDY bit of PERI_CSR0) and to clear RXPKTRDY by setting the SERV_RXPKTRDY bit of PERI_CSR0 (bit 6).

When the software detects the termination of a transfer (by receiving either the expected amount of data or an empty data packet), it should set the DATAEND bit (bit 3 of PERI_CSR0) to indicate to the controller that the data phase is complete and that the core should receive an acknowledge packet next.

Figure 15-7. Flow Chart of Receive Data Stage of a Control Transfer in Peripheral Mode

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.3.8.1.1.4.5 Error Handling: Control Transfer of Peripheral Mode

A control transfer may be aborted due to a protocol error on the USB, the host prematurely ending the transfer, or if the software wishes to abort the transfer (for example, because it cannot process the command).

The controller automatically detects protocol errors and sends a STALL packet to the host under the following conditions:

- The host sends more data during the OUT Data phase of a write request than was specified in the command. This condition is detected when the host sends an OUT token after the DATAEND bit (bit 3 of PERI_CSR0) has been set.
- The host requests more data during the IN Data phase of a read request than was specified in the command. This condition is detected when the host sends an IN token after the DATAEND bit in the PERI_CSR0 register has been set.
- The host sends more than Max Packet Size data bytes in an OUT data packet.
- The host sends a non-zero length DATA1 packet during the STATUS phase of a read request.

When the controller has sent the STALL packet, it sets the SENTSTALL bit (bit 2 of PERI_CSR0) and generates an interrupt. When the software receives an endpoint 0 interrupt with the SENTSTALL bit set, it should abort the current transfer, clear the SENTSTALL bit, and return to the IDLE state.

If the host prematurely ends a transfer by entering the STATUS phase before all the data for the request has been transferred, or by sending a new SETUP packet before completing the current transfer, then the SETUPEND bit (bit 4 of PERI_CSR0) will be set and an endpoint 0 interrupt generated. When the software receives an endpoint 0 interrupt with the SETUPEND bit set, it should abort the current transfer, set the SERV_SETUPEND bit (bit 7 of PERI_CSR0), and return to the IDLE state. If the RXPKTRDY bit (bit 0 of PERI_CSR0) is set this indicates that the host has sent another SETUP packet and the software should then process this command.

If the software wants to abort the current transfer, because it cannot process the command or has some other internal error, then it should set the SENDSTALL bit (bit 5 of PERI_CSR0). The controller will then send a STALL packet to the host, set the SENTSTALL bit (bit 2 of PERI_CSR0) and generate an endpoint 0 interrupt.

15.3.8.1.1.5 Additional Conditions: Control Transfer of Peripheral Mode

When working as a peripheral device, the controller automatically responds to certain conditions on the USB bus or actions by the host. The details are:

- Stall Issued to Control Transfers
  1. The host sends more data during an OUT Data phase of a Control transfer than was specified in the device request during the SETUP phase. This condition is detected by the controller when the host sends an OUT token (instead of an IN token) after the software has unloaded the last OUT packet and set DATAEND.
  2. The host requests more data during an IN data phase of a Control transfer than was specified in the device request during the SETUP phase. This condition is detected by the controller when the host sends an IN token (instead of an OUT token) after the software has cleared TXPKTRDY and set DATAEND in response to the ACK issued by the host to what should have been the last packet.
  3. The host sends more than MaxPktSize data with an OUT data token.
  4. The host sends the wrong PID for the OUT Status phase of a Control transfer.
  5. The host sends more than a zero length data packet for the OUT Status phase.
- Zero Length Out Data Packets In Control Transfer
  A zero length OUT data packet is used to indicate the end of a Control transfer. In normal operation, such packets should only be received after the entire length of the device request has been transferred (i.e., after the software has set DataEnd). If, however, the host sends a zero length OUT data packet before the entire length of device request has been transferred, this signals the premature end of the transfer. In this case, the controller will automatically flush any IN token loaded by software ready for the Data phase from the FIFO and set SETUPEND bit (bit 4 of PERI_CSR0).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.3.8.1.2 Bulk Transfer: Peripheral Mode

Bulk transactions are handled by endpoints other than endpoint 0. It is used to handle non-periodic, large bursty communication typically used for a transfer that use any available bandwidth and can also be delayed until bandwidth is available.

15.3.8.1.2.1 Bulk IN Transactions: Peripheral Mode

A Bulk IN transaction is used to transfer non-periodic data from the USB peripheral device to the host.

The following optional features are available for use with a Tx endpoint used in peripheral mode for Bulk IN transactions:

- Double packet buffering: When enabled, up to two packets can be stored in the FIFO awaiting transmission to the host. Double packet buffering is enabled by setting the DPB bit of TXFIFOSZ register (bit 4).
- DMA: If DMA is enabled for the endpoint, a DMA request will be generated whenever the endpoint is able to accept another packet in its FIFO. This feature allows the DMA controller to load packets into the FIFO without processor intervention

15.3.8.1.2.1.1 Bulk IN Transaction Setup: Peripheral Mode

In configuring a TX endpoint for bulk transactions, the TXMAXP register must be written with the maximum packet size (in bytes) for the endpoint. This value should be the same as the wMaxPacketSize field of the Standard Endpoint Descriptor for the endpoint and the PERI_TXCSR register (DMAEN and DMAMODE bit fields should be set when using DMA. Table 15-4 displays the PERI_TXCSR setting when used for Bulk transfer.

Table 15-4. PERI_TXCSR Register Bit Configuration for Bulk IN Transactions

Bit Field	Bit Name	Description
Bit 15	AUTOSET	Cleared to 0 if using DMA. For CPU Mode use, if AUTOSET bit is set, the TXPKTRDY bit will be automatically set when data of the maximum packet size is loaded into the FIFO.
Bit 14	ISO	Cleared to 0 for bulk mode operation.
Bit 13	MODE	Set to 1 to make sure the FIFO is enabled (only necessary if the FIFO is shared with an RX endpoint)
Bit 12	DMAEN	Set to 1 to enable DMA usage; not needed if CPU is being used to service the Tx Endpoint
Bit 11	FRCDATATOG	Cleared to 0 to allow normal data toggle operations.
Bit 10	DMAMODE	Set to 1 when DMA is used to service Tx FIFO.

When the endpoint is first configured (following a SET_CONFIGURATION or SET_INTERFACE command on Endpoint 0), the lower byte of PERI_TXCSR should be written to set the CLRDATATOG bit (bit 6). This will ensure that the data toggle (which is handled automatically by the controller) starts in the correct state.

Also if there are any data packets in the FIFO, indicated by the FIFONOTEMPTY bit (bit 1 of PERI_TXCSR) being set, they should be flushed by setting the FLUSHFIFO bit (bit 3 of PERI_TXCSR).

NOTE: It may be necessary to set this bit twice in succession if double buffering is enabled.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.3.8.1.2.1.2 Bulk IN Operation: Peripheral Mode

When data is to be transferred over a Bulk IN pipe, a data packet needs to be loaded into the FIFO and the PERI_TXCSR register written to set the TXPKTRDY bit (bit 0). When the packet has been sent, the TXPKTRDY bit will be cleared by the USB controller and an interrupt generated to signify to the user/application that the next packet can be loaded into the FIFO. If double packet buffering is enabled, then after the first packet has been loaded and the TXPKTRDY bit set, the TXPKTRDY bit will immediately be cleared (will not wait for the loaded first packet to be driven out) by the USB controller and an interrupt generated so that a second packet can be loaded into the FIFO. The software should operate in the same way, loading a packet when it receives an interrupt, regardless of whether double packet buffering is enabled or not.

In the general case, the packet size must not exceed the size specified by the lower 11 bits of the TXMAXP register. This part of the register defines the payload (packet size) for transfers over the USB and is required by the USB Specification to be either 8, 16, 32, 64 (Full-Speed or High-Speed) or 512 bytes (High-Speed only).

The host may determine that all the data for a transfer has been sent by knowing the total amount of data that is expected. Alternatively it may infer that all the data has been sent when it receives a packet which is smaller than the maximum packet size configuration for that particular endpoint (TXMAXP[10-0]). In the latter case, if the total size of the data block is a multiple of this payload, it will be necessary for the function to send a null packet after all the data has been sent. This is done by setting TXPKTRDY when the next interrupt is received, without loading any data into the FIFO.

If large blocks of data are being transferred, then the overhead of calling an interrupt service routine to load each packet can be avoided by using the CPPI DMA. A separate section detailing the use of the DMA is discussed within a latter section. Suffix is to say that the PERI_TXCSR (DMAEN and DMAMODE) bit fields need to be set and the PERI_TXCSR[AUTOSET] bit cleared at the core level when using the DMA with an endpoint configured for any IN transaction/transfer.

15.3.8.1.2.1.3 Error Handling of Bulk IN Transfer: Peripheral Mode

If the software wants to shut down the Bulk IN pipe, it should set the SENDSTALL bit (bit 4 of PERI_TXCSR). When the controller receives the next IN token, it will send a STALL to the host, set the SENTSTALL bit (bit 5 of PERI_TXCSR) and generate an interrupt.

When the software receives an interrupt with the SENTSTALL bit (bit 5 of PERI_TXCSR) set, it should clear the SENTSTALL bit. It should however leave the SENDSTALL bit set until it is ready to re-enable the Bulk IN pipe.

NOTE: If the host failed to receive the STALL packet for some reason, it will send another IN token, so it is advisable to leave the SENDSTALL bit set until the software is ready to re-enable the Bulk IN pipe. When a pipe is re-enabled, the data toggle sequence should be restarted by setting the CLRDATATOG bit in the PERI_TXCSR register (bit 6).

15.3.8.1.2.2 Bulk OUT Transfer: Peripheral Mode

A Bulk OUT transaction is used to transfer non-periodic data from the host to the function controller.

The following optional features are available for use with an Rx endpoint used in peripheral mode for Bulk OUT transactions:

- Double packet buffering: When enabled, up to two packets can be stored in the FIFO on reception from the host. Double packet buffering is enabled by setting the DPB bit of the RXFIFOSZ register (bit 4).
- DMA: If DMA is enabled for the endpoint, a DMA request will be generated whenever the endpoint has a packet in its FIFO. This feature can be used to allow the DMA controller to unload packets from the FIFO without processor intervention.

NOTE: When DMA is enabled, endpoint interrupt will not be generated for completion of packet reception. Endpoint interrupt will be generated only in the error conditions.

Copyright © 2011, Texas Instruments Incorporated

15.3.8.1.2.2.1 Bulk OUT Transfer Setup: Peripheral Mode

In configuring an Rx endpoint for Bulk OUT transactions, the RXMAXP register must be written with the maximum packet size (in bytes) for the endpoint. This value should be the same as the wMaxPacketSize field of the Standard Endpoint Descriptor for the endpoint. When using DMA for Rx Endpoints, the PERI_RXCSR needs to have its DMAEN bit set and have its AUTOCLEAR and DMAMODE bits cleared. In addition, the relevant interrupt enable bit in the INTRRXE register should be set (if an interrupt is required for this endpoint) and the PERI_RXCSR register should be set as shown in Table 15-5.

Table 15-5. PERI_RXCSR Register Bit Configuration for Bulk OUT Transactions

Bit Field	Bit Name	Description
Bit 15	AUTOCLEAR	Cleared to 0 if using DMA. In CPU Mode of usage, if the CPU sets AUTOCLEAR bit, the RXPKTRDY bit will be automatically cleared when a packet of RXMAXP bytes has been unloaded from the Receive FIFO.
Bit 14	ISO	Cleared to 0 to enable Bulk protocol.
Bit 13	DMAEN	Set to 1 if a DMA request is required for this Rx endpoint.
Bit 12	DISNYET	Cleared to 0 to allow normal PING flow control. This will affect only high speed transactions.
Bit 11	DMAMODE	Always clear this bit to 0.

When the Rx endpoint is first configured (following a SET_CONFIGURATION or SET_INTERFACE command on Endpoint 0), the lower byte of PERI_RXCSR should be written to set the CLRDATATOG bit (bit 7). This will ensure that the data toggle (which is handled automatically by the USB controller) starts in the correct state.

Also if there are any data packets in the FIFO (indicated by the RXPKTRDY bit (bit 0 of PERI_RXCSR) being set), they should be flushed by setting the FLUSHFIFO bit (bit 4 of PERI_RXCSR).

NOTE: It may be necessary to set this bit twice in succession if double buffering is enabled.

15.3.8.1.2.2.2 Bulk OUT Operation: Peripheral Mode

When a data packet is received by a Bulk Rx endpoint, the RXPKTRDY bit (bit 0 of PERI_RXCSR) is set and an interrupt is generated. The software should read the RXCOUNT register for the endpoint to determine the size of the data packet. The data packet should be read from the FIFO, and then the RXPKTRDY bit should be cleared.

The packets received should not exceed the size specified in the RXMAXP register (as this should be the value set in the wMaxPacketSize field of the endpoint descriptor sent to the host). When a block of data larger than wMaxPacketSize needs to be sent to the function, it will be sent as multiple packets. All the packets will be wMaxPacketSize in size, except the last packet which will contain the residue. The software may use an application specific method of determining the total size of the block and hence when the last packet has been received. Alternatively it may infer that the entire block has been received when it receives a packet which is less than wMaxPacketSize in size. (If the total size of the data block is a multiple of wMaxPacketSize, a null data packet will be sent after the data to signify that the transfer is complete.)

In the general case, the application software will need to read each packet from the FIFO individually. If large blocks of data are being transferred, the overhead of calling an interrupt service routine to unload each packet can be avoided by using DMA.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.3.8.1.2.2.3 Bulk OUT Error Handling: Peripheral Mode

If the software wants to shut down the Bulk OUT pipe, it should set the SENDSTALL bit (bit 5 of PERI_RXCSR). When the controller receives the next packet it will send a STALL to the host, set the SENTSTALL bit (bit 6 of PERI_RXCSR) and generate an interrupt.

When the software receives an interrupt with the SENTSTALL bit (bit 6 of PERI_RXCSR) set, it should clear this bit. It should however leave the SENDSTALL bit set until it is ready to re-enable the Bulk OUT pipe.

NOTE: If the host failed to receive the STALL packet for some reason, it will send another packet, so it is advisable to leave the SENDSTALL bit set until the software is ready to re-enable the Bulk OUT pipe. When a Bulk OUT pipe is re-enabled, the data toggle sequence should be restarted by setting the CLRDATATOG bit (bit 7) in the PERI_RXCSR register.

15.3.8.1.3 Interrupt Transfer: Peripheral Mode

An Interrupt IN transaction uses the same protocol as a Bulk IN transaction and the configuration and operation mentioned on prior sections apply to Interrupt transfer too with minor changes. Similarly, an Interrupt OUT transaction uses almost the same protocol as a Bulk OUT transaction and can be used the same way.

Tx endpoints in the USB controller have one feature for Interrupt IN transactions that they do not support in Bulk IN transactions. In Interrupt IN transactions, the endpoints support continuous toggle of the data toggle bit.

This feature is enabled by setting the FRCDATATOG bit in the PERI_TXCSR register (bit 11). When this bit is set, the controller will consider the packet as having been successfully sent and toggle the data bit for the endpoint, regardless of whether an ACK was received from the host.

Another difference is that interrupt endpoints do not support PING flow control. This means that the controller should never respond with a NYET handshake, only ACK/NAK/STALL. To ensure this, the DISNYET bit in the PERI_RXCSR register (bit 12) should be set to disable the transmission of NYET handshakes in high-speed mode.

Though DMA can be used with an interrupt OUT endpoint, it generally offers little benefit as interrupt endpoints are usually expected to transfer all their data in a single packet.

15.3.8.1.4 Isochronous Transfer: Peripheral Mode

Isochronous transfers are used when working with isochronous data. Isochronous transfers provide periodic, continuous communication between host and device.

15.3.8.1.4.1 Isochronous IN Transactions: Peripheral Mode

An Isochronous IN transaction is used to transfer periodic data from the function controller to the host.

The following optional features are available for use with a Tx endpoint used in Peripheral mode for Isochronous IN transactions:

- Double packet buffering: When enabled, up to two packets can be stored in the FIFO awaiting transmission to the host. Double packet buffering is enabled by setting the DPB bit of TXFIFOSZ register (bit 4).

  NOTE: Double packet buffering is generally advisable for isochronous transactions in order to avoid underrun errors as described in later section.
- DMA: If DMA is enabled for the endpoint, a DMA request will be generated whenever the endpoint is able to accept another packet in its FIFO. This feature allows the DMA controller to load packets into the FIFO without processor intervention.
  However, this feature is not particularly useful with Isochronous endpoints because the packets transferred are often not maximum packet size and the PERI_TXCSR register needs to be accessed following every packet to check for Underrun errors.
  When DMA is enabled and DMAMODE bit of PERI_TXCSR is set, endpoint interrupt will not be generated for completion of packet transfer. Endpoint interrupt will be generated only in the error conditions.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.3.8.1.4.1.1 Isochronous IN Transfer Setup: Peripheral Mode

In configuring a Tx endpoint for Isochronous IN transactions, the TXMAXP register must be written with the maximum packet size (in bytes) for the endpoint. This value should be the same as the wMaxPacketSize field of the Standard Endpoint Descriptor for the endpoint. In addition, the relevant interrupt enable bit in the INTRTXE register should be set (if an interrupt is required for this endpoint) and the PERI_TXCSR register should be set as shown in Table 15-6.

Table 15-6. PERI_TXCSR Register Bit Configuration for Isochronous IN Transactions

Bit Field	Bit Name	Description
Bit 14	ISO	Set to 1 to enable Isochronous transfer protocol.
Bit 13	MODE	Set to 1 to ensure the FIFO is enabled (only necessary if the FIFO is shared with an Rx endpoint).
Bit 12	DMAEN	Set to 1 if DMA Requests have to be enabled.
Bit 11	FRCDATATOG	Ignored in Isochronous mode.
Bit 10	DMAMODE	Set to 1 when DMA is enabled.

15.3.8.1.4.1.2 Isochronous IN Transfer Operation: Peripheral Mode

An isochronous endpoint does not support data retries, so if data underrun is to be avoided, the data to be sent to the host must be loaded into the FIFO before the IN token is received. The host will send one IN token per frame (or microframe in High-speed mode), however the timing within the frame (or microframe) can vary. If an IN token is received near the end of one frame and then at the start of the next frame, there will be little time to reload the FIFO. For this reason, double buffering of the endpoint is usually necessary.

An interrupt is generated whenever a packet is sent to the host and the software may use this interrupt to load the next packet into the FIFO and set the TXPKTRDY bit in the PERI_TXCSR register (bit 0) in the same way as for a Bulk Tx endpoint. As the interrupt could occur almost any time within a frame(/microframe), depending on when the host has scheduled the transaction, this may result in irregular timing of FIFO load requests. If the data source for the endpoint is coming from some external hardware, it may be more convenient to wait until the end of each frame(/microframe) before loading the FIFO as this will minimize the requirement for additional buffering. This can be done by using either the SOF interrupt or the external SOF_PULSE signal from the controller to trigger the loading of the next data packet. The SOF_PULSE is generated once per frame(/microframe) when a SOF packet is received. (The controller also maintains an external frame(/microframe) counter so it can still generate a SOF_PULSE when the SOF packet has been lost.) The interrupts may still be used to set the TXPKTRDY bit in PERI_TXCSR (bit 0) and to check for data overruns/underruns.

Starting up a double-buffered Isochronous IN pipe can be a source of problems. Double buffering requires that a data packet is not transmitted until the frame(/microframe) after it is loaded. There is no problem if the function loads the first data packet at least a frame(/microframe) before the host sets up the pipe (and therefore starts sending IN tokens). But if the host has already started sending IN tokens by the time the first packet is loaded, the packet may be transmitted in the same frame(/microframe) as it is loaded, depending on whether it is loaded before, or after, the IN token is received. This potential problem can be avoided by setting the ISOUPDATE bit in the POWER register (bit 7). When this bit is set, any data packet loaded into an Isochronous Tx endpoint FIFO will not be transmitted until after the next SOF packet has been received, thereby ensuring that the data packet is not sent too early.

15.3.8.1.4.1.3 Isochronous IN Error Handling: Peripheral Mode

If the endpoint has no data in its FIFO when an IN token is received, it will send a null data packet to the host and set the UNDERRUN bit in the PERI_TXCSR register (bit 2). This is an indication that the software is not supplying data fast enough for the host. It is up to the application to determine how this error condition is handled.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

If the software is loading one packet per frame(/microframe) and it finds that the TXPKTRDY bit in the PERI_TXCSR register (bit 0) is set when it wants to load the next packet, this indicates that a data packet has not been sent (perhaps because an IN token from the host was corrupted). It is up to the application how it handles this condition: it may choose to flush the unsent packet by setting the FLUSHFIFO bit in the PERI_TXCSR register (bit 3), or it may choose to skip the current packet.

15.3.8.1.4.2 Isochronous OUT Transactions: Peripheral Mode

An isochronous OUT transaction is used to transfer periodic data from the host to the function controller.

Following optional features are available for use with an Rx endpoint used in Peripheral mode for Isochronous OUT transactions:

- Double packet buffering: When enabled, up to two packets can be stored in the FIFO on reception from the host. Double packet buffering is enabled by setting the DPB bit of RXFIFOSZ register (bit 4). **NOTE:** Double packet buffering is generally advisable for Isochronous transactions in order to avoid overrun errors.
- DMA: If DMA is enabled for the endpoint, a DMA request will be generated whenever the endpoint has a packet in its FIFO. This feature can be used to allow the DMA controller to unload packets from the FIFO without processor intervention.
  However, this feature is not particularly useful with Isochronous endpoints because the packets transferred are often not maximum packet size and the PERI_RXCSR register needs to be accessed following every packet to check for Overrun or CRC errors.
  When DMA is enabled, endpoint interrupt will not be generated for completion of packet reception. Endpoint interrupt will be generated only in the error conditions.

15.3.8.1.4.2.1 Isochronous OUT Setup: Peripheral Mode

In configuring an Rx endpoint for Isochronous OUT transactions, the RXMAXP register must be written with the maximum packet size (in bytes) for the endpoint. This value should be the same as the wMaxPacketSize field of the Standard Endpoint Descriptor for the endpoint. In addition, the relevant interrupt enable bit in the INTRRXE register should be set (if an interrupt is required for this endpoint) and the PERI_RXCSR register should be set as shown in Table 15-7.

Table 15-7. PERI_RXCSR Register Bit Configuration for Isochronous OUT Transactions

Bit Field	Bit Name	Description
Bit 15	AUTOCLEAR	Cleared to 0 if using DMA. In CPU Mode of usage, if the CPU sets AUTOCLEAR bit, the RXPKTRDY bit will be automatically cleared when a packet of RXMAXP bytes has been unloaded from the Receive FIFO.
Bit 14	ISO	Set to 1 to configure endpoint usage for Isochronous transfer.
Bit 13	DMAEN	Set to 1 if a DMA request is required for this Rx endpoint.
Bit 12	DISNYET	Ignored in Isochronous Mode.
Bit 11	DMAMODE	Always clear this bit to 0.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.3.8.1.4.2.2 Isochronous OUT Operation: Peripheral Mode

An Isochronous endpoint does not support data retries so, if a data overrun is to be avoided, there must be space in the FIFO to accept a packet when it is received. The host will send one packet per frame (or microframe in High-speed mode); however, the time within the frame can vary. If a packet is received near the end of one frame(/microframe) and another arrives at the start of the next frame, there will be little time to unload the FIFO. For this reason, double buffering of the endpoint is usually necessary.

An interrupt is generated whenever a packet is received from the host and the software may use this interrupt to unload the packet from the FIFO and clear the RXPKTRDY bit in the PERI_RXCSR register (bit 0) in the same way as for a Bulk Rx endpoint. As the interrupt could occur almost any time within a frame(/microframe), depending on when the host has scheduled the transaction, the timing of FIFO unload requests will probably be irregular. If the data sink for the endpoint is going to some external hardware, it may be better to minimize the requirement for additional buffering by waiting until the end of each frame(/microframe) before unloading the FIFO. This can be done by using either the SOF interrupt or the external SOF_PULSE signal from the controller to trigger the unloading of the data packet. The SOF_PULSE is generated once per frame(/microframe) when a SOF packet is received. (The controller also maintains an external frame(/microframe) counter so it can still generate a SOF_PULSE when the SOF packet has been lost.) The interrupts may still be used to clear the RXPKTRDY bit in PERI_RXCSR and to check for data overruns/underruns.

15.3.8.1.4.2.3 Isochronous OUT Error Handling: Peripheral Mode

If there is no space in the FIFO to store a packet when it is received from the host, the OVERRUN bit in the PERI_RXCSR register (bit 2) will be set. This is an indication that the software is not unloading data fast enough for the host. It is up to the application to determine how this error condition is handled

If the controller finds that a received packet has a CRC error, it will still store the packet in the FIFO and set the RXPKTRDY bit (bit 0 of PERI_RXCSR) and the DATAERROR bit (bit 3 of PERI_RXCSR). It is left up to the application to determine how this error condition is handled.

The number of USB packets sent in any microframe will depend on the amount of data to be transferred, and is indicated through the PIDs used for the individual packets. If the indicated number of packets have not been received by the end of a microframe, the INCOMPRX bit (bit 8) bit in the PERI_RXCSR register will be set to indicate that the data in the FIFO is incomplete. Equally, if a packet of the wrong data type is received, then the PID Error bit is the PERI_RXCSR register will be set. In each case, an interrupt will, however, still be generated to allow the data that has been received to be read from the FIFO.

Note: The circumstances in which a PID Error or INCOMPRX is reported depends on the precise sequence of packets received.

When the core is operating in peripheral mode, the details are in Table 15-8.

Table 15-8. Isochronous OUT Error Handling: Peripheral Mode

No. Packet(s) Expected	Data Packet(s) Received	Response
1	DATA0	OK
	DATA1	PID Error set
	DATA2	PID Error set
	MDATA	PID Error set
2	DATA0	OK
	DATA1	INCOMPRX Set
	DATA2	INCOMPRX Set + PID Error set
	MDATA	INCOMPRX Set
	MDATA DATA0	PID Error Set
	MDATA DATA1	OK
	MDATA DATA2	PID Error Set
	MDATA MDATA	PID Error Set
3	DATA0	OK
	DATA1	INCOMPRX Set

Copyright © 2011, Texas Instruments Incorporated

Table 15-8. Isochronous OUT Error Handling: Peripheral Mode (continued)

No. Packet(s) Expected	Data Packet(s) Received	Response
	DATA2	INCOMPRX Set
	MDATA	INCOMPRX Set
	MDATA DATA0	PID Error set
	MDATA DATA1	OK
	MDATA DATA2	INCOMPRX Set
	MDATA MDATA	INCOMPRX Set
	MDATA MDATA DATA0	PID Error set
	MDATA MDATA DATA1	PID Error set
	MDATA MDATA DATA2	OK
	MDATA MDATA MDATA	PID Error set

15.3.8.2 USB Controller Host Mode Operation

The USB controller assumes the role of a host when the USBx_ID pin state is grounded or USB Mode Register[iddig=bit8] is cleared to 0 (provided that iddig_mux, which is bit7 of USBnMODE is also set to 1) by the user application prior to the controller goes into session. When the USB controller go into session, application/firmware sets the DEVCTL[SESSION] bit to 1, it will assume the role of a host.

- Entry into Suspend mode. When operating as a host, the USB controller can be prompted to go into Suspend mode by setting the SUSPENDM bit in the POWER register. When this bit is set, the controller will complete the current transaction then stop the transaction scheduler and frame counter. No further transactions will be started and no SOF packets will be generated.
  If the ENSUSPM bit (bit 0 of POWER register) is set, the UTMI+ PHY will go into low-power mode when the controller goes into suspend mode.
- Sending Resume Signaling. When the application requires the controller to leave suspend mode, it needs to clear the SUSPENDM bit in the POWER register, set the RESUME bit and leave it set for 20ms. While the RESUME bit is high, the controller will generate Resume signaling on the bus. After 20 ms, the CPU should clear the RESUME bit, at which point the frame counter and transaction scheduler will be started.
- Responding to Remote Wake-up. If Resume signaling is detected from the target while the controller is in suspend mode, the UTMI+ PHY will be brought out of low-power mode and UTMI clock is restarted. The controller will then exit suspend mode and automatically set the RESUME bit in the POWER register (bit 2) to ‘1’ to take over generating the Resume signaling from the target. If the resume interrupt is enabled, an interrupt will be generated.
- Reset Signaling. If the RESET bit in the POWER register (bit 3) is set while the controller is in Host mode, it will generate Reset signaling on the bus. If the HSENAB bit in the POWER register (bit 5) was set, it will also try to negotiate for high-speed operation. The software should keep the RESET bit set for at least 20 ms to ensure correct resetting of the target device. After the software has cleared the bit, the controller will start its frame counter and transaction scheduler. Whether high-speed operation is selected will be indicated by HSMODE bit of POWER register (bit 4).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.3.8.2.1 Control Transactions: Host Mode

Host control transactions are conducted through Endpoint 0 and the software is required to handle all the Standard Device Requests that may be sent or received via Endpoint 0 (as described in Universal Serial Bus Specification, Revision 2.0). Endpoint 0 can only be serviced via CPU and DMA mode can not be used.

There are three categories of Standard Device Requests to be handled: Zero Data Requests (in which all the information is included in the command); Write Requests (in which the command will be followed by additional data); and Read Requests (in which the device is required to send data back to the host).

1. Zero Data Requests comprise a SETUP command followed by an IN Status Phase.
2. Write Requests comprise a SETUP command, followed by an OUT Data Phase which is in turn followed by an IN Status Phase.
3. Read Requests comprise a SETUP command, followed by an IN Data Phase which is in turn followed by an OUT Status Phase.

A timeout may be set to limit the length of time for which the host controller will retry a transaction which is continually NAKed by the target. This limit can be between 2 and 2 15 frames/microframes and is set through the NAKLIMIT0 register.

The following sections describe the actions that the CPU needs to take in issuing these different types of request through looking at the steps to take in the different phases of a control transaction.

Note: Before initiating any transactions as a Host, the FADDR register needs to be set to address the peripheral device. When the device is first connected, FADDR should be set to zero. After a SET_ADDRESS command is issued, FADDR should be set the target's new address.

15.3.8.2.1.1 Setup Phase of Control Transaction: Host Mode

For the SETUP Phase of a control transaction (Figure 15-8), the software driving the USB host device needs to:

1. Load the 8 bytes of the required Device request command into the Endpoint 0 FIFO
2. Set SETUPPKT and TXPKTRDY (bits 3 and 1 of HOST_CSR0, respectively).
   NOTE: These bits must be set together.
   The controller then proceeds to send a SETUP token followed by the 8-byte command/request to Endpoint 0 of the addressed device, retrying as necessary. *Note:* On errors, the controller retries the transaction three times.
3. At the end of the attempt to send the 8-byte request data, the controller will generate an Endpoint 0 interrupt. The software should then read HOST_CSR0 to establish whether the RXSTALL bit (bit 2), the ERROR bit (bit 4) or the NAK_TIMEOUT bit (bit 7) has been set.
   If RXSTALL is set, it indicates that the target did not accept the command (for example, because it is not supported by the target device) and so has issued a STALL response. If ERROR is set, it means that the controller has tried to send the SETUP Packet and the following data packet three times without getting any response.
   If NAK_TIMEOUT is set, it means that the controller has received a NAK response to each attempt to send the SETUP packet, for longer than the time set in HOST_NAKLIMIT0. The controller can then be directed either to continue trying this transaction (until it times out again) by clearing the NAK_TIMEOUT bit or to abort the transaction by flushing the FIFO before clearing the NAK_TIMEOUT bit.
4. If none of RXSTALL, ERROR or NAK_TIMEOUT is set, the SETUP Phase has been correctly ACKed and the software should proceed to the following IN Data Phase, OUT Data Phase or IN Status Phase specified for the particular Standard Device Request.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 15-8. Flow Chart of Setup Stage of a Control Transfer in Host Mode

DRAFT ONLY

15.3.8.2.1.2 Data Phase (IN Data Phase) of a Control Transaction: Host Mode

For the IN Data Phase of a control transaction (Figure 15-9), the software driving the USB host device needs to

1. Set REQPKT bit of HOST_CSR0 (bit 5)
2. Wait while the controller sends the IN token and receives the required data back.
3. When the controller generates the Endpoint 0 interrupt, read HOST_CSR0 to establish whether the RXSTALL bit (bit 2), the ERROR bit (bit 4), the NAK_TIMEOUT bit (bit 7) or RXPKTRDY bit (bit 0) has been set.
   If RXSTALL is set, it indicates that the target has issued a STALL response.

Copyright © 2011, Texas Instruments Incorporated

If ERROR is set, it means that the controller has tried to send the required IN token three times without getting any response. If NAK_TIMEOUT bit is set, it means that the controller has received a NAK response to each attempt to send the IN token, for longer than the time set in HOST_NAKLIMIT0. The controller can then be directed either to continue trying this transaction (until it times out again) by clearing the NAK_TIMEOUT bit or to abort the transaction by clearing REQPKT before clearing the NAK_TIMEOUT bit

4. If RXPKTRDY has been set, the software should read the data from the Endpoint 0 FIFO, then clear RXPKTRDY.
5. If further data is expected, the software should repeat Steps 1-4.

When all the data has been successfully received, the CPU should proceed to the OUT Status Phase of the Control Transaction.

Figure 15-9. Flow Chart of Data Stage (IN Data Phase) of a Control Transfer in Host Mode

15.3.8.2.1.3 Data Phase (OUT Data Phase) of a Control Transaction: Host Mode

For the OUT Data Phase of a control transaction (Figure 15-10), the software driving the USB host device needs to:

Copyright © 2011, Texas Instruments Incorporated

1. Load the data to be sent into the endpoint 0 FIFO.
2. Set the TXPKTRDY bit of HOST_CSR0 (bit 1). The controller then proceeds to send an OUT token followed by the data from the FIFO to Endpoint 0 of the addressed device, retrying as necessary.
3. At the end of the attempt to send the data, the controller will generate an Endpoint 0 interrupt. The software should then read HOST_CSR0 to establish whether the RXSTALL bit (bit 2), the ERROR bit (bit 4) or the NAK_TIMEOUT bit (bit 7) has been set.
   If RXSTALL bit is set, it indicates that the target has issued a STALL response.
   If ERROR bit is set, it means that the controller has tried to send the OUT token and the following data packet three times without getting any response. If NAK_TIMEOUT is set, it means that the controller has received a NAK response to each attempt to send the OUT token, for longer than the time set in the HOST_NAKLIMIT0 register. The controller can then be directed either to continue trying this transaction (until it times out again) by clearing the NAK_TIMEOUT bit or to abort the transaction by flushing the FIFO before clearing the NAK_TIMEOUT bit.
   If none of RXSTALL, ERROR or NAKLIMIT is set, the OUT data has been correctly ACKed.
4. If further data needs to be sent, the software should repeat Steps 1-3.
   When all the data has been successfully sent, the software should proceed to the IN Status Phase of the Control Transaction.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 15-10. Flow Chart of Data Stage (OUT Data Phase) of a Control Transfer in Host Mode

DRAFT ONLY

15.3.8.2.1.4 Status Phase (Status for OUT Data Phase or Setup Phase) of a Control Transaction: Host Mode

IN Status Phase of a control transfer exists for a Zero Data Request or for a Write Request of a control transfer. The IN Status Phase follows the Setup Stage, if no Data Stage of a control transfer exists, or OUT Data Phase of a Data Stage of a control transfer.

For the IN Status Phase of a control transaction (Figure 15-11), the software driving the USB Host device needs to:

1. Set the STATUSPKT and REQPKT bits of HOST_CSR0 (bit 6 and bit 5, respectively).
2. Wait while the controller sends an IN token and receives a response from the USB peripheral device.
3. When the controller generates the Endpoint 0 interrupt, read HOST_CSR0 to establish whether the RXSTALL bit (bit 2), the ERROR bit (bit 4), the NAK_TIMEOUT bit (bit 7) or RXPKTRDY bit (bit 0) has been set.

Copyright © 2011, Texas Instruments Incorporated

If RXSTALL bit is set, it indicates that the target could not complete the command and so has issued a STALL response.
If ERROR bit is set, it means that the controller has tried to send the required IN token three times without getting any response.
If NAK_TIMEOUT bit is set, it means that the controller has received a NAK response to each attempt to send the IN token, for longer than the time set in the HOST_NAKLIMIT0 register. The controller can then be directed either to continue trying this transaction (until it times out again) by clearing the NAK_TIMEOUT bit or to abort the transaction by clearing REQPKT bit and STATUSPKT bit before clearing the NAK_TIMEOUT bit.

4. The CPU should clear the STATUSPPKT bit of HOST_CSR0 together with (i.e., in the same write operation as) RxPktRdy if this has been set.

Figure 15-11. Flow Chart of Status Stage of Zero Data Request or Write Request of a Control Transfer in Host Mode

15.3.8.2.1.5 Status Phase for a Read Request of a Control Transaction: Host Mode

OUT Status Phase of a control transfer exist for a Read Request or a control transfer where data was received by the host controller. The OUT Status phase follows the IN Data Stage of a control transfer.

Copyright © 2011, Texas Instruments Incorporated

For the OUT Status Phase of a control transaction (Figure 15-12), the CPU driving the host device needs to:

1. Set STATUSPKT and TXPKTRDY bits of HOST_CSR0 (bit 6 and bit 1, respectively).
   NOTE: These bits need to be set together
2. Wait while the controller sends the OUT token and a zero-length DATA1 packet.
3. At the end of the attempt to send the data, the controller will generate an Endpoint 0 interrupt. The software should then read HOST_CSR0 to establish whether the RXSTALL bit (bit 2), the ERROR bit (bit 4) or the NAK_TIMEOUT bit (bit 7) has been set.
   If RXSTALL bit is set, it indicates that the target could not complete the command and so has issued a STALL response.
   If ERROR bit is set, it means that the controller has tried to send the STATUS Packet and the following data packet three times without getting any response.
   If NAK_TIMEOUT bit is set, it means that the controller has received a NAK response to each attempt to send the IN token, for longer than the time set in the HOST_NAKLIMIT0 register. The controller can then be directed either to continue trying this transaction (until it times out again) by clearing the NAK_TIMEOUT bit or to abort the transaction by flushing the FIFO before clearing the NAK_TIMEOUT bit.
4. If none of RXSTALL, ERROR or NAK_TIMEOUT bits is set, the STATUS Phase has been correctly ACKed.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 15-12. Chart of Status Stage of a Read Request of a Control Transfer in Host Mode

DRAFT ONLY

15.3.8.2.2 Bulk Transfer: Host Mode

Bulk transactions are handled by endpoints other than endpoint 0. It is used to handle non-periodic, large bursty communication typically used for a transfer that use any available bandwidth and can also be delayed until bandwidth is available.

15.3.8.2.2.1 Bulk IN Transactions: Host Mode

A Bulk IN transaction may be used to transfer non-periodic data from the external USB peripheral to the host.

Copyright © 2011, Texas Instruments Incorporated

The following optional features are available for use with an Rx endpoint used in host mode to receive the data:

- Double packet buffering: When enabled, up to two packets can be stored in the FIFO on reception from the host. This allows that one packet can be received while another is being read. Double packet buffering is enabled by setting the DPB bit of RXFIFOSZ register (bit 4).
- DMA: If DMA is enabled for the endpoint, a DMA request will be generated whenever the endpoint has a packet in its FIFO. This feature can be used to allow the DMA controller to unload packets from the FIFO without processor intervention.
  When DMA is enabled, endpoint interrupt will not be generated for completion of packet reception. Endpoint interrupt will be generated only in the error conditions. For more information see section on CPPI DMA

15.3.8.2.2.1.1 Bulk IN Setup: Host Mode

Before initiating any Bulk IN Transactions in Host mode:

- The HOST_RXTYPE register for the endpoint that is to be used needs to be programmed as:
  - Operating speed in the SPEED bit field (bits 7 and 6).
  - Set 10 (binary value) in the PROT field for bulk transfer.
  - Endpoint Number of the target device in RENDPN field. This is the endpoint number contained in the Rx endpoint descriptor returned by the target device during enumeration.
- The RXMAXP register for the controller endpoint must be written with the maximum packet size (in bytes) for the transfer. This value should be the same as the wMaxPacketSize field of the Standard Endpoint Descriptor for the target endpoint.
- The HOST_RXINTERVAL register needs to be written with the required value for the NAK limit (2-215 frames/microframes), or set to zero if the NAK timeout feature is not required.
- The relevant interrupt enable bit in the INTRRXE register should be set (if an interrupt is required for this endpoint).

  Note: Whether interrupt is handled from PDR level or Core level, interrupt is required to be enabled at the core level. See details on Core Interrupt registers, CTRLR register, and sections on Interrupt Handling
- The AUTOREQ bit field should be used only when servicing transfers using CPU mode and must be cleared when using DMA Mode. For DMA Mode, dedicated registers USB0/1 Req Registers exist and the HOST_RXCSR[AUTOREQ] should be cleared.
  When DMA is enabled, the following bits of HOST_RXCSR register should be set as:
  - Clear AUTOCLEAR.
  - Set DMAEN (bit 13) to 1 if a DMA request is required for this endpoint.
  - Clear DSINYET (bit 12) to 0 to allow normal PING flow control. This will affect only High Speed transactions.
  - Clear DMAMODE (bit 11) to 0.

    Note: If DMA is enabled, the USB0/1 Auto Req register can be set for generating IN tokens automatically after receiving the data. Set the bit field RXn_AUTOREQ (where n is the endpoint number) with binary value 01 or 11.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

When the endpoint is first configured, the endpoint data toggle should be cleared to 0 either by using the DATATOGWREN and DATATOG bits of HOST_RXCSR (bit 10 and bit 9) to toggle the current setting or by setting the CLRDATATOG bit of HOST_RXCSR (bit 7). This will ensure that the data toggle (which is handled automatically by the controller) starts in the correct state. Also if there are any data packets in the FIFO (indicated by the RXPKTRDY bit (bit 0 of HOST_RXCSR) being set), they should be flushed by setting the FLUSHFIFO bit of HOST_RXCSR (bit 4).

NOTE: It may be necessary to set this bit twice in succession if double buffering is enabled.

15.3.8.2.2.1.2 Bulk IN Operation: Host Mode

When Bulk data is required from the USB peripheral device, the software should set the REQPKT bit in the corresponding HOST_RXCSR register (bit 5). The controller will then send an IN token to the selected peripheral endpoint and waits for data to be returned.

If data is correctly received, RXPKTRDY bit of HOST_RXCSR (bit 0) is set. If the USB peripheral device responds with a STALL, RXSTALL bit (bit 6 of HOST_RXCSR) is set. If a NAK is received, the controller tries again and continues to try until either the transaction is successful or the POLINTVL_NAKLIMIT set in the HOST_RXINTERVAL register is reached. If no response at all is received, two further attempts are made before the controller reports an error by setting the ERROR bit of HOST_RXCSR (bit 2).

The controller then generates the appropriate endpoint interrupt, whereupon the software should read the corresponding HOST_RXCSR register to determine whether the RXPKTRDY, RXSTALL, ERROR or DATAERR_NAKTIMEOUT bit is set and act accordingly. If the DATAERR_NAKTIMEOUT bit is set, the controller can be directed either to continue trying this transaction (until it times out again) by clearing the DATAERR_NAKTIMEOUT bit or to abort the transaction by clearing REQPKT bit before clearing the DATAERR_NAKTIMEOUT bit.

The packets received should not exceed the size specified in the RXMAXP register (as this should be the value set in the wMaxPacketSize field of the endpoint descriptor sent to the host).

In the general case, the application software (if CPU is servicing the endpoint) will need to read each packet from the FIFO individually. If large blocks of data are being transferred, the overhead of calling an interrupt service routine to unload each packet can be avoided by using DMA.

Note: When using DMA, see CPPI DMA section for the proper configuration of the core register, HOST_RXCSR.

15.3.8.2.2.1.3 Bulk IN Error Handling: Host Mode

If the target wants to shut down the Bulk IN pipe, it will send a STALL response to the IN token. This will result in the RXSTALL bit of HOST_RXCSR (bit 6) being set.

15.3.8.2.2.2 Bulk OUT Transactions: Host Mode

A Bulk OUT transaction may be used to transfer non-periodic data from the host to the USB peripheral.

Following optional features are available for use with a Tx endpoint used in Host mode to transmit this data:

- Double packet buffering: When enabled, up to two packets can be stored in the FIFO awaiting transmission to the peripheral device. Double packet buffering is enabled by setting the DPB bit of TXFIFOSZ register (bit 4).
- DMA: If DMA is enabled for the endpoint, a DMA request will be generated whenever the endpoint is able to accept another packet in its FIFO. This feature can be used to allow the DMA controller to load packets into the FIFO without processor intervention. For more information on using DMA, consult the section discussing CPPI DMA.
  When DMA is enabled and DMAMODE bit in HOST_TXCSR register is set, an endpoint interrupt will not be generated for completion of packet reception. An endpoint interrupt will be generated only in the error conditions.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.3.8.2.2.2.1 Bulk OUT Setup: Host Mode

Before initiating any bulk OUT transactions:

- The target function address needs to be set in the TXFUNCADDR register for the selected controller endpoint. (TXFUNCADDR register is available for all endpoints from EP0 to EP15.)
- The HOST_TXTYPE register for the endpoint that is to be used needs to be programmed as:
  - Operating speed in the SPEED bit field (bits 7 and 6).
  - Set PROT field to 10b for bulk transfer.
  - Enter Endpoint Number of the target device in TENDPN field. This is the endpoint number contained in the OUT(Tx) endpoint descriptor returned by the target device during enumeration.
- The TXMAXP register for the controller endpoint must be written with the maximum packet size (in bytes) for the transfer. This value should be the same as the wMaxPacketSize field of the Standard Endpoint Descriptor for the target endpoint.
- The HOST_TXINTERVAL register needs to be written with the required value for the NAK limit (2-215 frames/microframes), or set to zero if the NAK timeout feature is not required.
- The relevant interrupt enable bit in the INTRTXE register should be set (if an interrupt is required for this endpoint).
- The following bits of HOST_TXCSR register should be set as:
  - Set the MODE bit (bit 13) to 1 to ensure the FIFO is enabled (only necessary if the FIFO is shared with an Rx endpoint).
  - Clear the FRCDATATOG bit (bit 11) to 0 to allow normal data toggle operations.
  - Clear/Set AUTOSET bit (bit 15). The setting of this bit depends on the desire of the user/application in automatically setting the TXPKTRDY bit when servicing transactions using CPU.

Note: If DMA is needs to be used in place of the CPU, the following table displays the setting of the core register HOST_TXCSR register in Host mode. For the CPPI DMA registers settings consult the section on CPPI DMA within this document.

Bit Field	Bit Name	Description
Bit 15	AUTOSET	Cleared to 0 if using DMA. For CPU Mode use, if AUTOSET bit is set, the TXPKTRDY bit will be automatically set when data of the maximum packet size is loaded into the FIFO.
Bit 14	ISO	Cleared to 0 for bulk mode operation.
Bit 13	MODE	Set to 1 to make sure the FIFO is enabled (only necessary if the FIFO is shared with an RX endpoint)
Bit 12	DMAEN	Set to 1 to enable DMA usage; not needed if CPU is being used to service the Tx Endpoint
Bit 11	FRCDATATOG	Cleared to 0 to allow normal data toggle operations.
Bit 10	DMAMODE	Set to 1 when DMA is used to service Tx FIFO.

When the endpoint is first configured, the endpoint data toggle should be cleared to 0 either by using the DATATOGWREN bit and DATATOG bit of HOST_TXCSR (bit 9 and bit 8) to toggle the current setting or by setting the CLRDATATOG bit of HOST_TXCSR (bit 6). This will ensure that the data toggle (which is handled automatically by the controller) starts in the correct state. Also, if there are any data packets in the FIFO (indicated by the FIFONOTEMPTY bit of HOST_TXCSR register (bit 1) being set), they should be flushed by setting the FLUSHFIFO bit (bit 3 of HOST_TXCSR).

NOTE: It may be necessary to set this bit twice in succession if double buffering is enabled.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.3.8.2.2.2.2 Bulk OUT Operation: Host Mode

When Bulk data is required to be sent to the USB peripheral device, the software should write the first packet of the data to the FIFO (or two packets if double-buffered) and set the TXPKTRDY bit in the corresponding HOST_TXCSR register (bit 0). The controller will then send an OUT token to the selected peripheral endpoint, followed by the first data packet from the FIFO.

If data is correctly received by the peripheral device, an ACK should be received whereupon the controller will clear TXPKTRDY bit of HOST_TXCSR (bit 0). If the USB peripheral device responds with a STALL, the RXSTALL bit (bit 5) of HOST_TXCSR is set. If a NAK is received, the controller tries again and continues to try until either the transaction is successful or the NAK limit set in the HOST_TXINTERVAL register is reached. If no response at all is received, two further attempts are made before the controller reports an error by setting ERROR bit in HOST_TXCSR (bit 2).

The controller then generates the appropriate endpoint interrupt, whereupon the software should read the corresponding HOST_TXCSR register to determine whether the RXSTALL (bit 5), ERROR (bit 2) or NAK_TIMEOUT (bit 7) bit is set and act accordingly. If the NAK_TIMEOUT bit is set, the controller can be directed either to continue trying this transaction (until it times out again) by clearing the NAK_TIMEOUT bit or to abort the transaction by flushing the FIFO before clearing the NAK_TIMEOUT bit.

If large blocks of data are being transferred, then the overhead of calling an interrupt service routine to load each packet can be avoided by using DMA.

15.3.8.2.2.2.3 Bulk OUT Error Handling: Host Mode

If the target wants to shut down the Bulk OUT pipe, it will send a STALL response. This is indicated by the RXSTALL bit of HOST_TXCSR register (bit 5) being set.

15.3.8.2.3 Interrupt Transfer: Host Mode

When the controller is operating as the host, interactions with an Interrupt endpoint on the USB peripheral device are handled in very much the same way as the equivalent Bulk transactions (described in previous sections).

The principal difference as far as operational steps are concerned is that the PROT field of HOST_RXTYPE and HOST_TXTYPE (bits 5-4) need to be set (binary value) to represent an Interrupt transaction. The required polling interval also needs to be set in the HOST_RXINTERVAL and HOST_TXINTERVAL registers.

15.3.8.2.4 Isochronous Transfer: Host Mode

Isochronous transfers are used when working with isochronous data. Isochronous transfers provide periodic, continuous communication between host and device.

An Isochronous IN transaction is used to transfer periodic data from the USB peripheral to the host.

15.3.8.2.4.1 Isochronous IN Transactions: Host Mode

The following optional features are available for use with an Rx endpoint used in Host mode to receive this data:

- Double packet buffering: When enabled, up to two packets can be stored in the FIFO on reception from the host. This allows that one packet can be received while another is being read. Double packet buffering is enabled by setting the DPB bit of RXFIFOSZ register (bit 4).
- AutoRequest: When the AutoRequest feature is enabled, the REQPKT bit of HOST_RXCSR (bit 5) will be automatically set when the RXPKTRDY bit is cleared. This only applies when the CPU is being used to service the endpoint. When using DMA, this bit field needs to be cleared to Zero. CPPI DMA has its own configuration registers that renders a similar task, USB0/1 Auto Req registers, that needs to be used to have a similar effect. For more information, see section on CPPI DMA.
- DMA: If DMA is enabled for the endpoint, a DMA request will be generated whenever the endpoint has a packet in its FIFO. This feature can be used to allow the DMA controller to unload packets from the FIFO without processor intervention. However, this feature is not particularly useful with isochronous endpoints because the packets transferred are often not maximum packet size.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

- When DMA is enabled, endpoint interrupt will not be generated for completion of packet reception. Endpoint interrupt will be generated only in the error conditions.

15.3.8.2.4.1.1 Isochronous IN Transfer Setup: Host

Before initiating an Isochronous IN Transactions in Host mode:

- The target function address needs to be set in the RXFUNCADDR register for the selected controller endpoint (RXFUNCADDR register is available for all endpoints from EP0 to EP4).
- The HOST_RXTYPE register for the endpoint that is to be used needs to be programmed as:
  - Operating speed in the SPEED bit field (bits 7 and 6).
  - Set 01 (binary value) in the PROT field for isochronous transfer.
  - Endpoint Number of the target device in RENDPN field. This is the endpoint number contained in the Rx endpoint descriptor returned by the target device during enumeration.
- The RXMAXP register for the controller endpoint must be written with the maximum packet size (in bytes) for the transfer. This value should be the same as the wMaxPacketSize field of the Standard Endpoint Descriptor for the target endpoint.
- The HOST_RXINTERVAL register needs to be written with the required transaction interval (usually one transaction per frame/microframe).
- The relevant interrupt enable bit in the INTRRXE register should be set (if an interrupt is required for this endpoint).
- The following bits of HOST_RXCSR register should be set as:
  - Clear AUTOCLEAR
  - Set the DMAEN bit (bit 13) to 1 if a DMA request is required for this endpoint.
  - Clear the DISNYET it (bit 12) to 0 to allow normal PING flow control. This will only affect High Speed transactions.
  - Clear DMAMODE bit (bit 11) to 0.
- If DMA is enabled, AUTOREQ register can be set for generating IN tokens automatically after receiving the data. Set the bit field RXn_AUTOREQ (where n is the endpoint number) with binary value 01 or 11. For detailed information on using CPPI DMA, consult related section within this document.

15.3.8.2.4.1.2 Isochronous IN Operation: Host Mode

The operation starts with the software setting REQPKT bit of HOST_RXCSR (bit 5). This causes the controller to send an IN token to the target.

When a packet is received, an interrupt is generated which the software may use to unload the packet from the FIFO and clear the RXPKTRDY bit in the HOST_RXCSR register (bit 0) in the same way as for a Bulk Rx endpoint. As the interrupt could occur almost any time within a frame(/microframe), the timing of FIFO unload requests will probably be irregular. If the data sink for the endpoint is going to some external hardware, it may be better to minimize the requirement for additional buffering by waiting until the end of each frame before unloading the FIFO. This can be done by using the SOF_PULSE signal from the controller to trigger the unloading of the data packet. The SOF_PULSE is generated once per frame(/microframe). The interrupts may still be used to clear the RXPKTRDY bit in HOST_RXCSR.

15.3.8.2.4.1.3 Isochronous IN Error Handling: Host Mode

If a CRC or bit-stuff error occurs during the reception of a packet, the packet will still be stored in the FIFO but the DATAERR_NAKTIMEOUT bit of HOST_RXCSR (bit 3) is set to indicate that the data may be corrupt.

Note: The number of USB packets sent in any microframe will depend on the amount of data to be transferred, and is indicated through the PIDs used for the individual packets. If the indicated numbers of packets have not been received by the end of a microframe, the INCOMPRX bit in the HOST_RXCSR register will be set to indicate that the data in the FIFO is incomplete. Equally, if a packet of the wrong data type is received, then the PID Error bit is the HOST_RXCSR register will be set. In each case, an interrupt will, however, still be generated to allow the data that has been received to be read from the FIFO.

Copyright © 2011, Texas Instruments Incorporated

15.3.8.2.4.2 Isochronous OUT Transactions: Host Mode

An Isochronous OUT transaction may be used to transfer periodic data from the host to the USB peripheral.

Following optional features are available for use with a Tx endpoint used in Host mode to transmit this data:

- Double packet buffering: When enabled, up to two packets can be stored in the FIFO awaiting transmission to the peripheral device. Double packet buffering is enabled by setting the DPB bit of TXFIFOSZ register (bit 4).

DMA: If DMA is enabled for the endpoint, a DMA request will be generated whenever the endpoint is able to accept another packet in its FIFO. This feature can be used to allow the DMA controller to load packets into the FIFO without processor intervention.

However, this feature is not particularly useful with isochronous endpoints because the packets transferred are often not maximum packet size.

When DMA is enabled and endpoint interrupt will not be generated for completion of packet reception. Endpoint interrupt will be generated only in the error conditions.

15.3.8.2.4.2.1 Isochronous OUT Transfer Setup: Host Mode

Before initiating any Isochronous OUT transactions:

- The target function address needs to be set in the TXFUNCADDR register for the selected controller endpoint (TXFUNCADDR register is available for all endpoints from EP0 to EP4).
- The HOST_TXTYPE register for the endpoint that is to be used needs to be programmed as:
  - Operating speed in the SPEED bit field (bits 7 and 6).
  - Set 01 (binary value) in the PROT field for isochronous transfer.
  - Endpoint Number of the target device in TENDPN field. This is the endpoint number contained in the OUT(Tx) endpoint descriptor returned by the target device during enumeration.
- The TXMAXP register for the controller endpoint must be written with the maximum packet size (in bytes) for the transfer. This value should be the same as the wMaxPacketSize field of the Standard Endpoint Descriptor for the target endpoint.
- The HOST_TXINTERVAL register needs to be written with the required transaction interval (usually one transaction per frame/microframe).
- The relevant interrupt enable bit in the INTRTXE register should be set (if an interrupt is required for this endpoint).
- The following bits of HOST_TXCSR register should be set as:
  - Set the MODE bit (bit 13) to 1 to ensure the FIFO is enabled (only necessary if the FIFO is shared with an Rx endpoint).
  - Set the DMAEN bit (bit 12) to 1 if a DMA request is required for this endpoint
  - The FRCDATATOG bit (bit 11) is ignored for isochronous transactions.
  - Set the DMAMODE bit (bit 10) to 1 when DMA is enabled.
    For more details in using DMA, consult CPPI DMA section within this document.

15.3.8.2.4.2.2 Isochronous OUT Transfer Operation: Host Mode

The operation starts when the software writes to the FIFO and sets TXPKTRDY bit of HOST_TXCSR (bit 0). This triggers the controller to send an OUT token followed by the first data packet from the FIFO.

An interrupt is generated whenever a packet is sent and the software may use this interrupt to load the next packet into the FIFO and set the TXPKTRDY bit in the HOST_TXCSR register (bit 0) in the same way as for a Bulk Tx endpoint. As the interrupt could occur almost any time within a frame, depending on when the host has scheduled the transaction, this may result in irregular timing of FIFO load requests. If

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

the data source for the endpoint is coming from some external hardware, it may be more convenient to wait until the end of each frame before loading the FIFO as this will minimize the requirement for additional buffering. This can be done by using the SOF_PULSE signal from the controller to trigger the loading of the next data packet. The SOF_PULSE is generated once per frame(/microframe). The interrupts may still be used to set the TXPKTRDY bit in HOST_TXCSR.

15.3.8.2.4.3 High Bandwidth Isochronous Transactions: Host Mode

Place Holder: Details to be populated.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.3.9 Communications Port Programming Interface (CPPI) 4.1 DMA

The CPPI DMA module supports the transmission and reception of USB packets. The CPPI DMA is designed to facilitate the segmentation and reassembly of CPPI compliant packets to/from smaller data blocks that are natively compatible with the specific requirements of each networking port. Multiple Tx and Rx channels are provided for all endpoints (excluding endpoint 0) within the DMA allowing multiple segmentation or reassembly operations to be effectively performed in parallel (but not actually simultaneously). The DMA controller maintains state information for each of the ports/channels which allows packet segmentation and reassembly operations to be time division multiplexed between channels in order to share the underlying DMA hardware. A DMA scheduler is used to control the ordering and rate at which this multiplexing occurs.

The CPPI (version 4.1) DMA controller sub-module is a common 15 port DMA controller. It supports 15 Tx and 15 Rx channels and each port attaches to the associated endpoint in the controller. Port 1 maps to endpoint 1 and Port 2 maps to endpoint 2 and so on with Port 15 mapped to endpoint 15; endpoint 0 can not utilize the DMA and the firmware is responsible to load or offload the endpoint 0 FIFO via CPU.

15.3.9.1 CPPI Terminology

Host — The host is an intelligent system resource that configures and manages each communications control module. The host is responsible for allocating memory, initializing all data structures, and responding to port interrupts.

Main Memory — The area of data storage managed by the CPU. The CPPI DMA (CDMA) reads and writes CPPI packets from and to main memory. This memory can exist internal or external from the device.

Queue Manager (QM) — The QM is responsible for accelerating management of a variety of Packet Queues and Free Descriptor / Buffer Queues. It provides status indications to the CDMA Scheduler when queues are empty or full.

CPPI DMA (CDMA) — The CDMA is responsible for transferring data between the CPPI FIFO and Main Memory. It acquires free Buffer Descriptor from the QM (Receive Submit Queue) for storage of received data, posts received packets pointers to the Receive Completion Queue, transmits packets stored on the Transmit Submit Queue (Transmit Queue) , and posts completed transmit packets to the Transmit Completion Queue.

CDMA Scheduler (CDMAS) — The CDMAS is responsible for scheduling CDMA transmit and receive operations. It uses Queue Indicators from the QM and the CDMA to determine the types of operations to schedule.

CPPI FIFO — The CPPI FIFO provides FIFO interfaces (for each of the 15 transmit and 15 receive endpoints).

Transfer DMA (XDMA) — The XDMA receives DMA requests from the Mentor USB 2.0 Core and initiates DMAs to the CPPI FIFO.

Endpoint FIFOs — The Endpoint FIFOs are the USB packet storage elements used by the Mentor USB 2.0 Core for packet transmission or reception. The XDMA transfers data between the CPPI FIFO and the Endpoint FIFOs for transmit operations and between the Endpoint FIFOs and the CPPI FIFO for receive operations.

Port — A port is the communications module (peripheral hardware) that contains the control logic for Direct Memory Access for a single transmit/receive interface or set of interfaces. Each port may have multiple communication channels that transfer data using homogenous or heterogeneous protocols. A port is usually subdivided into transmit and receive pairs which are independent of each other. Each endpoint, excluding endpoint 0, has its own dedicated port.

Channel — A channel refers to the sub-division of information (flows) that is transported across ports. Each channel has associated state information. Channels are used to segregate information flows based on the protocol used, scheduling requirements (example: CBR, VBR, ABR), or concurrency requirements (that is, blocking avoidance). All fifteen ports per USB Module have dedicated single channels, channel 0, associated for their use in a USB application.

Copyright © 2011, Texas Instruments Incorporated

Data Buffer — A data buffer is a single data structure that contains payload information for transmission to or reception from a port. A data buffer is a byte aligned contiguous block of memory used to store packet payload data. A data buffer may hold any portion of a packet and may be linked together (via descriptors) with other buffers to form packets. Data buffers may be allocated anywhere within the 32-bit memory space. The Buffer Length field of the packet descriptor indicates the number of valid data bytes in the buffer. There may be from 1 to 4M-1 valid data bytes in each buffer.

Host Buffer Descriptor (Buffer Descriptor) — A buffer descriptor is a single data structure that contains information about one or more data buffers. This type of descriptor is required when more than one descriptor is needed to define an entire packet, i.e., it either defines the middle of a packet or end of a packet.

Start of Packet (SOP) — Packet along side Buffer Descriptors are used to hold information about Packets. When multiple Descriptors are used to hold a single packet information or a single Descriptor is used to hold a single packet information, the first descriptor is referred to as a Packet Descriptor which is also Start-of-Packet Descriptor.

Host Packet Descriptor (Packet Descriptor) — A packet descriptor is another name for the first buffer descriptor within a packet. Some fields within a data buffer descriptor are only valid when it is a packet descriptor including the tags, packet length, packet type, and flags. This type of descriptor is always used to define a packet since it provides packet level information that is useful to both the ports and the Host in order to properly process the packet. It is the only descriptor used when single descriptor solely defines a packet. When multiple descriptors are needed to define a packet, the packet descriptor is the first descriptor used to define a packet.

Free Descriptor/Buffer Queue — A free descriptor/buffer queue is a hardware managed list of available descriptors with pre-linked empty buffers that are to be used by the receive ports for host type descriptors. Free Descriptor/Buffer Queues are implemented by the Queue Manager.

Teardown Descriptor — Teardown Descriptor is a special structure which is not used to describe either a packet or a buffer but is instead used to describe the completion of a channel halt and teardown event. Channel teardown is an important function because it ensures that when a connection is no longer needed that the hardware can be reliably halted and any remaining packets which had not yet been transmitted can be reclaimed by the Host without the possibility of losing buffer or descriptor references (which results in a memory leak).

Packet Queue — A packet queue is hardware managed list of valid (i.e. populated) packet descriptors that is used for forwarding a packet from one entity to another for any number of purposes.
NOTE: All descriptors (regardless of type) must be allocated at addresses that are naturally aligned to the smallest power of 2 that is equal to or greater than the descriptor size.

15.3.9.2 Data Structures

Two data structures are mainly used to identify data buffers used by packet and buffer descriptors. A third Descriptor, Teardown Descriptor, exists. The purpose of this Descriptor is a channel halt and teardown event. Each of these Descriptor layouts as well as bit fields are shown below.

15.3.9.2.1 Host Packet Descriptor/ Packet Descriptor (SOP Descriptor)

Descriptors are designed to be used when USB like application requires support for true, unlimited fragment count scatter/gather type operations. The Packet Descriptor is the first descriptor on multiple descriptors setup or the only descriptor in a single descriptors setup. The Packet Descriptor contains the following information:

- Indicator which identifies the descriptor as a packet descriptor (always 10h
- Source and destination tags (reserved)
- Packet type
- Packet length
- Protocol specific region size
- Protocol specific control/statusbits
- Pointer to the first valid byte in the SOP data buffer

Copyright © 2011, Texas Instruments Incorporated

- Length of the SOP data buffer
- Pointer to the next buffer descriptor in the

Packet descriptors can vary in size by design of their defined fields from 32 bytes up to 104 bytes. Within this range, packet descriptors always contain 32 bytes of required information and may also contain 8 bytes of software specific tagging information and up to 64 bytes (indicated in 4 byte increments) of protocol specific information. How much protocol specific information (and therefore the allocated size of the descriptors) is application dependent. Port will make use of the first 32 bytes only.

From a general USB use perspective, a 32-byte descriptor size is suffix and the use of this size is expected for a normal USB usage.

The packet descriptor layout is shown in Figure 15-13 and described within Table 15-9 through Table 15-16.

Figure 15-13. Packet Descriptor Layout

Table 15-9. Packet Descriptor Word 0 (PD0) Bit Field Descriptions

Bits	Field Name	Description
31-27	Descriptor type	The host packet descriptor type is 16 decimal (10h). The CPU initializes this field.
26-22	Protocol-specific valid word count	This field indicates the valid number of 32-bit words in the protocol-specific region. The CPU initializes this field. This is encoded in increments of 4 bytes as: 0 = 0 byte 1 = 4 bytes ... 16 = 64 bytes 17-31 = Reserved
21-0	Packet length	The length of the packet in bytes. If the packet length is less than the sum of the buffer lengths, then the packet data will be truncated. A packet length greater than the sum of the buffers is an error. The valid range for the packet length is 0 to (4M - 1) bytes. The CPU initializes this field for transmitted packets; the DMA overwrites this field on packet reception.

Copyright © 2011, Texas Instruments Incorporated

Table 15-10. Packet Descriptor Word 1 (PD1) Bit Field Descriptions

Bits	Field Name	Description
31-27	Source Tag; Port #	This field indicates the port number (0-31) from which the packet originated. The DMA overwrites this field on packet reception. This is the RX Endpoint number from which the packet originated.
26-21	Source Tag; Channel #	This field indicates the channel number within the port from which the packet originated. The DMA overwrites this field on packet reception. This field is always 0-67.
20-16	Source Tag: Sub-channel #	This field indicates the sub-channel number (0-31) within the channel from which the packet originated. The DMA overwrites this field on packet reception. This field is always 0.
15-0	Destination Tag	This field is application-specific. This field is always 0.

Table 15-11. Packet Descriptor Word 2 (PD2) Bit Field Descriptions

Bits	Field Name	Description
31	Packet error	This bit indicates if an error occurred during reception of this packet (0 = no error occurred; 1 = error occurred). The DMA overwrites t his field on packet reception. Additional information about different errors may be encoded in the protocol specific fields in the descriptor.
30-26		This field indicates the type of this packet. The CPU initializes this field for transmitted packets; the DMA overwrites this field on packet reception. This field is encoded as: 5 = USB 8-31 = Reserved
25-20	Reserved	Reserved
19	Zero-length packet indicator	If a zero-length USB packet is received, the XDMA will send the CDMA a data block with a byte count of 0 and this bit is set. The CDMA will then perform normal EOP termination of the packet without transferring data. For transmit, if a packet has this bit set, the XDMA will ignore the CPPI packet size and send a zero-length packet to the USB controller.
18-16	Protocol-specific	This field contains protocol-specific flags/information that can be assigned based on the packet type. Not used for USB.
15	Return policy	This field indicates the return policy for this packet. The CPU initializes this field. 0 = Entire packet (still linked together) should be returned to the queue specified in bits 13-0. 1 = Each buffer should be returned to the queue specified in bits 13-0 of Word 2 in their respective descriptors. The Tx DMA will return each buffer in sequence.
14	On-chip	This field indicates whether or not this descriptor is in a region which is in on-chip memory space (1) or in external memory (0).
31-12	Packet return queue mgr #	This field indicates which queue manager in the system the descriptor is to be returned to after transmission is complete. This field is not altered by the DMA during transmissionor reception and is initialized by the CPU. There is only one queue manager in the USB HS/FS device controller. This field must always be 0.
11-0	Packet return queue #	This field indicates the queue number within the selected queue manager that the descriptor is to be returned to after transmission is complete. This field is not altered by the DMA during transmission or reception and is initialized by the CPU.

Table 15-12. Packet Descriptor Word 3 (PD3) Bit Field Descriptions

Bits	Field Name	Description
31-22	Reserved	Reserved
21-0	Buffer 0 length	The buffer length field indicates how many valid data bytes are in the buffer. The CPU initializes this field for transmitted packets; the DMA overwrites this field on packet reception.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-13. Packet Descriptor Word 4 (PD4) Bit Field Descriptions

Bits	Field Name	Description
31-0	Buffer 0 pointer	The buffer pointer is the byte-aligned memory address of the buffer associated with the buffer descriptor. The CPU initializes this field for transmitted packets; the DMA overwrites this field on packet reception.

Table 15-14. Packet Descriptor Word 5 (PD5) Bit Field Descriptions

Bits	Field Name	Description
31-0	Next descriptor pointer	The 32-bit word aligned memory address of the next buffer descriptor in the packet. If the value of this pointer is zero, then the current buffer is the last buffer in the packet. The CPU initializes this field for transmitted packets; the DMA overwrites this field on packet reception.

Table 15-15. Packet Descriptor Word 6 (PD6) Bit Field Descriptions

Bits	Field Name	Description
31-22	Reserved	Reserved
21-0	Original buffer 0 length	The buffer length field indicates the original size of the buffer in bytes. This value is not overwritten during reception. This value is read by the Rx DMA to determine the actual buffer size as allocated by the CPU at initialization. Since the buffer length in Word 3 is overwritten by the Rx port during reception, this field is necessary to permanently store the buffer size information.

Table 15-16. Packet Descriptor Word 7 (PD7) Bit Field Descriptions

Bits	Field Name	Description
31-22	Reserved	Reserved
21-0	Original buffer 0 pointer	The buffer pointer is the byte-aligned memory address of the buffer associated with the buffer descriptor. This value is not overwritten during reception. This value is read by the Rx DMA to determine the actual buffer location as allocated by the CPU at initialization. Since the buffer pointer in Word 4 is overwritten by the Rx port during reception, this field is necessary to permanently store the buffer pointer information.

15.3.9.2.2 Host Buffer Descriptor/Buffer Descriptor (BD)

The buffer descriptor is identical in size and organization to a Packet Descriptor but does not include valid information in the packet level fields and does not include a populated region for protocol specific information. The packet level fields are not needed since the SOP descriptor contains this information and additional copy of this data is not needed/necessary.

Host buffer descriptors are designed to be linked onto a host packet descriptor or another host buffer descriptor to provide support for unlimited scatter / gather type operations. Host buffer descriptors provide information about a single corresponding data buffer. Every host buffer descriptor stores the following information:

- Pointer to the first valid byte in the data
- Length of the data buffer
- Pointer to the next buffer descriptor in the packet

Buffer descriptors always contain 32 bytes of required information. Since it is a requirement that it is possible to convert a descriptor between a Buffer Descriptor and a Packet Descriptor (by filling in the appropriate fields) in practice, Buffer Descriptors will be allocated using the same sizes as Packet Descriptors. In addition, since the 5 LSBs of the Descriptor Pointers are used in CPPI 4.1 for the purpose of indicating the length of the descriptor, the minimum size of a descriptor is always 32 bytes

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

The buffer descriptor layout is shown in Figure 15-14 and described within Table 15-17 through Table 15-24 .

Figure 15-14. Buffer Descriptor (BD) Layout

Table 15-17. Buffer Descriptor Word 0 (BD0) Bit Field Descriptions

Bits	Field Name	Description
31-0	Reserved	Reserved

Table 15-18. Buffer Descriptor Word 1 (BD1) Bit Field Descriptions

Bits	Field Name	Description
31-0	Reserved	Reserved

Table 15-19. Buffer Descriptor Word 2 (BD2) Bit Field Descriptions

Bits	Field Name	Description
31-15	Reserved	Reserved
14	On-chip	This field indicates whether or not this descriptor is in a region which is on-chip memory space (1) or in external memory (0).
13-12	Packet return queue mgr #	This field indicates which queue manager in the system the descriptor is to be returned to after transmission is complete. This field is not altered by the DMA during transmission or reception and is initialized by the CPU. There is only 1 queue manager in the USB HS/FS device controller. This field must always be 0.
11-0	Packet return queue #	This field indicates the queue number within the selected queue manager that the descriptor is to be returned to after transmission is complete. This field is not altered by the DMA during transmission or reception and is initialized by the CPU.

Table 15-20. Buffer Descriptor Word 3 (BD3) Bit Field Descriptions

Bits	Field Name	Description
31-22	Reserved	Reserved
21-0	Buffer 0 length	The buffer length field indicates how many valid data bytes are in the buffer. The CPU initializes this field for transmitted packets. The DMA overwrites this field upon packet reception.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-21. Buffer Descriptor Word 4 (BD4) Bit Field Descriptions

Bits	Field Name	Description
31-0	Buffer 0 pointer	The buffer pointer is the byte-aligned memory address of the buffer associated with the buffer descriptor. The CPU initializes this field for transmitted packets. The DMA overwrites this field upon packet reception.

Table 15-22. Buffer Descriptor Word 5 (BD5) Bit Field Descriptions

Bits	Field Name	Description
31-0	Next description pointer	The 32-bit word aligned memory address of the next buffer descriptor in the packet. If the value of this pointer is zero, then the current descriptor is the last descriptor in the packet. The CPU initializes this field for transmitted packets. The DMA overwrites this field upon packet reception.

Table 15-23. Buffer Descriptor Word 6 (BD6) Bit Field Descriptions

Bits	Field Name	Description
31-22	Reserved	Reserved
21-0	Original buffer 0 length	The buffer length field indicates the original size of the buffer in bytes. This value is not overwritten during reception. This value is read by the Rx DMA to determine the actual buffer size as allocated by the CPU at initialization. Since the buffer length in Word 3 is overwritten by the Rx port during reception, this field is necessary to permanently store the buffer size information.

Table 15-24. Buffer Descriptor Word 7 (BD7) Bit Field Descriptions

Bits	Field Name	Description
31-0	Original buffer 0 pointer	The buffer pointer is the byte-aligned memory address of the buffer associated with the buffer descriptor. This value is not overwritten during reception. This value is read by the Rx DMA to determine the actual buffer location as allocated by the CPU at initialization. Since the buffer pointer in Word 4 is overwritten by the Rx port during reception, this field is necessary to permanently store the buffer pointer information.

15.3.9.2.3 Teardown Descriptor

The Teardown Descriptor is not like the packet or buffer descriptors since it is not used to describe either a packet or a buffer. The teardown descriptor is always 32 bytes long and is comprised of 4 bytes of actual teardown information and 28 bytes of pad. The teardown descriptor layout and associated bit field descriptions are shown in the Figure below and Tables that follows. Since the 5 LSBs of the descriptor pointers are used in CPPI 4.1 for the purpose of indicating the length of the descriptor, the minimum size of a descriptor is 32 bytes.

The teardown descriptor is used to describe a channel halt and teardown event. Channel teardown ensures that when a connection is no longer needed that the hardware can be reliably halted and any remaining packets which had not yet been transmitted can be reclaimed by the host without the possibility of losing buffer or descriptor references (which results in a memory leak).

Theteardown descriptor contains the following information:

- Indicator which identifies the descriptor as a teardown packet descriptor
- DMA controller number where teardown
- Channel number within DMA where teardown occurred
- Indicator of whether this teardown was for the Tx or Rx channel

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 15-15. Teardown Descriptor Layout

Table 15-25. Teardown Descriptor Word 0 Bit Field Descriptions

Bits	Field Name	Description
31-27	Descriptor type	The teardown descriptor type is 19 decimal (13h)
26-17	Reserved	Reserved
16	TX_RX	Indicates whether teardown is a TX (0) or RX (1).
15-10	DMA number	Indicates the DMA number for this teardown.
9-6	Reserved	Reserved
5-0	Channel number	Indicates the channel number within the DMA that was torn down.

Table 15-26. Teardown Descriptor Words 1 to 7 Bit Field Descriptions

Bits	Field Name	Description
31-0	Reserved	Reserved

Teardown operation of an endpoint requires three operations. The teardown register in the CPPI DMA must be written, the corresponding endpoint bit in TEARDOWN of the USB module must be set, and the FlushFIFO bit in the Mentor USB controller Tx/RxCSR register must be set.

The following is the Transmit teardown procedure highlighting the steps required to be followed:

1. Set the TX_TEARDOWN bit in the CPPI DMA TX channel n global configuration register (TXGCRn).
2. Set the appropriate TX_TDOWN bit in the USBOTG controller's USB teardown register (TEARDOWN). Write Tx Endpoint Number to teardown to TEARDOWN[TX_TDOWN] field.
3. Check if the teardown descriptor has been received on the teardown queue: The completion queue (see Queue Assignment table) is usually used as the Teardown queue when the Teardown descriptor has been received, the descriptor address will be loaded onto CTRLD[Completion Queue #] register:
   (a) If not, go to step 2
   (b) If so, go to step 4
4. Set the appropriate TX_TDOWN bit in the USBOTG controller's USB teardown register (TEARDOWN). Set the bit corresponding to the Channel Number within TEARDOWN[TX_TDOWN] field.
5. Flush the TX FIFO in the Mentor OTG core: Set PERI_TXCSR[FLUSHFIFO] for the corresponding Endpoint.
6. Re-enable the Tx DMA channel.
   (a) Clear TXGCRn[TX_TEARDOWN and TX_ENABLE] bit.
   (b) Set TXGCRn[TX_ENABLE] bit.

Copyright © 2011, Texas Instruments Incorporated

15.3.9.3 Queue Manager

The queue manager (QM) is a hardware module that is responsible for accelerating management of the queues, i.e. queues are maintained within a queue manager module. Packets are added to a queue by writing the 32-bit descriptor address to a particular memory mapped location in the queue manager module. Packets are de-queued by reading the same location for that particular queue. A single queue manager exists within the device and handles all available 156 queues within the USB subsystem

Descriptors are queued onto a logical queue (pushed) by writing the descriptor pointer to the Queue N Register D. When the Queue N Register D is written, this kicks off the queue manager causing it to add the descriptor to the tail of queue.

The QM keeps track of the order of the descriptors pushed within a queue or the linking status of the descriptors. To accomplish this linking status, QM will first resolve the 32-bit descriptor pointer into a 16-bit index which is used for linking a queue pointer purposes. Once the address to index computation is performed, i.e., the physical index information is determined, the QM will link that descriptor onto the descriptor chain that is maintained for that logical queue by writing the linking information out to a linking RAM which is external to the queue manager. More on linking RAM discussion would follow in latter sections. The QM will also update the queue tail pointers appropriately. Since logical queues within the queue manager are maintained using linked lists, queues cannot become full and no check for fullness is required before queuing a packet descriptor.

15.3.9.3.1 Queue Types

Several types of queues exist (a total of 156 queues) within the CPPI 4.1 DMA. Regardless of the type of queue, queues are used to hold pointers to Packet or Buffer Descriptors while they are being passed between the Host and / or any of the ports in the system. All queues are maintained within the single Queue Manager module.

The following types of Queues exist:

- Free descriptor queue (unassigned to specific endpoint but assigned to specific endpoint type; receive endpoints – can be used as a receive submit queue)
- Transmit submit queue
- Transmit completion (return) queue
- Receive completion (return) queue
- Teardown queue

Dedicated queues exist where one or more queues are assigned for the use of a single endpoint and non-dedicated queues exist where no specific queue assignment to an endpoint is required (but pertains to receive endpoints only). Transmit endpoints are not allowed to use free descriptor queues.

Dedicated queues exist for each specific endpoint use and non-dedicated queues exist that can be used by any/all receive endpoints. Three queues are reserved for each endpoint/port for transmit actions with two of the three queues being transmit submit queues while the remaining queue is used as a completion/return queue. For receive actions, the only dedicated queues are completion/return queues; one queue is assigned/reserved for each receive endpoint. 32 free descriptor queues that do not have dedicated endpoint-queue assignment exist and these queues are used to service any receive endpoint as receive submit queues.

Table 15-27 displays queue-endpoint assignments.

Table 15-27. Queue-Endpoint Assignments

Queue Start Number	Queue Count	Queue-Endpoint Association
0	32	Free Descriptor Queues/Rx Submit Queues (USB0/1 Rx Endpoints)
32	2	USB0 Tx Endpoint 1
34	2	USB0 Tx Endpoint 2
36	2	USB0 Tx Endpoint 3
38	2	USB0 Tx Endpoint 4

Copyright © 2011, Texas Instruments Incorporated

Table 15-27. Queue-Endpoint Assignments (continued)

Queue Start Number	Queue Count	Queue-Endpoint Association
40	2	USB0 Tx Endpoint 5
42	2	USB0 Tx Endpoint 6
44	2	USB0 Tx Endpoint 7
46	2	USB0 Tx Endpoint 8
48	2	USB0 Tx Endpoint 9
50	2	USB0 Tx Endpoint 10
52	2	USB0 Tx Endpoint 11
54	2	USB0 Tx Endpoint 12
56	2	USB0 Tx Endpoint 13
58	2	USB0 Tx Endpoint 14
60	2	USB0 Tx Endpoint 15
62	2	USB1 Tx Endpoint 1
64	2	USB1 Tx Endpoint 2
66	2	USB1 Tx Endpoint 3
68	2	USB1 Tx Endpoint 4
70	2	USB1 Tx Endpoint 5
72	2	USB1 Tx Endpoint 6
74	2	USB1 Tx Endpoint 7
76	2	USB1 Tx Endpoint 8
78	2	USB1 Tx Endpoint 9
80	2	USB1 Tx Endpoint 10
82	2	USB1 Tx Endpoint 11
84	2	USB1 Tx Endpoint 12
86	2	USB1 Tx Endpoint 13
88	2	USB1 Tx Endpoint 14
90	2	USB1 Tx Endpoint 15
92	1	Reserved
93	1	USB0 Tx Endpoint 1 completion queue
94	1	USB0 Tx Endpoint 2 completion queue
95	1	USB0 Tx Endpoint 3 completion queue
96	1	USB0 Tx Endpoint 4 completion queue
97	1	USB0 Tx Endpoint 5 completion queue
98	1	USB0 Tx Endpoint 6 completion queue
99	1	USB0 Tx Endpoint 7 completion queue
100	1	USB0 Tx Endpoint 8 completion queue
101	1	USB0 Tx Endpoint 9 completion queue
102	1	USB0 Tx Endpoint 10 completion queue
103	1	USB0 Tx Endpoint 11 completion queue
104	1	USB0 Tx Endpoint 12 completion queue
104	1	USB0 Tx Endpoint 13 completion queue
106	1	USB0 Tx Endpoint 14 completion queue
107	1	USB0 Tx Endpoint 15 completion queue
108	1	Reserved
109	1	USB0 Rx Endpoint 1 completion queue
110	1	USB0 Rx Endpoint 2 completion queue
111	1	USB0 Rx Endpoint 3 completion queue
112	1	USB0 Rx Endpoint 4 completion queue

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-27. Queue-Endpoint Assignments (continued)

Queue Start Number	Queue Count	Queue-Endpoint Association
113	1	USB0 Rx Endpoint 5 completion queue
114	1	USB0 Rx Endpoint 6 completion queue
115	1	USB0 Rx Endpoint 7 completion queue
116	1	USB0 Rx Endpoint 8 completion queue
117	1	USB0 Rx Endpoint 9 completion queue
118	1	USB0 Rx Endpoint 10 completion queue
119	1	USB0 Rx Endpoint 11 completion queue
120	1	USB0 Rx Endpoint 12 completion queue
121	1	USB0 Rx Endpoint 13 completion queue
122	1	USB0 Rx Endpoint 14 completion queue
123	1	USB0 Rx Endpoint 15 completion queue
124	1	Reserved
125	1	USB1 Tx Endpoint 1 completion queue
126	1	USB1 Tx Endpoint 2 completion queue
127	1	USB1 Tx Endpoint 3 completion queue
128	1	USB1 Tx Endpoint 4 completion queue
129	1	USB1 Tx Endpoint 5 completion queue
130	1	USB1 Tx Endpoint 6 completion queue
131	1	USB1 Tx Endpoint 7 completion queue
132	1	USB1 Tx Endpoint 8 completion queue
133	1	USB1 Tx Endpoint 9 completion queue
134	1	USB1 Tx Endpoint 10 completion queue
135	1	USB1 Tx Endpoint 11 completion queue
136	1	USB1 Tx Endpoint 12 completion queue
137	1	USB1 Tx Endpoint 13 completion queue
138	1	USB1 Tx Endpoint 14 completion queue
139	1	USB1 Tx Endpoint 15 completion queue
140	1	Reserved
141	1	USB1 Rx Endpoint 1 completion queue
142	1	USB1 Rx Endpoint 2 completion queue
143	1	USB1 Rx Endpoint 3 completion queue
144	1	USB1 Rx Endpoint 4 completion queue
145	1	USB1 Rx Endpoint 5 completion queue
146	1	USB1 Rx Endpoint 6 completion queue
147	1	USB1 Rx Endpoint 7 completion queue
148	1	USB1 Rx Endpoint 8 completion queue
149	1	USB1 Rx Endpoint 9 completion queue
150	1	USB1 Rx Endpoint 10 completion queue
151	1	USB1 Rx Endpoint 11 completion queue
152	1	USB1 Rx Endpoint 12 completion queue
153	1	USB1 Rx Endpoint 13 completion queue
154	1	USB1 Rx Endpoint 14 completion queue
155	1	USB1 Rx Endpoint 15 completion queue

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.3.9.3.2 Free Descriptor Queue (Receive Submit Queue)

Receive endpoints/channels use queues, referred to as free descriptor queues, or receive submit queues, to forward completed receive packets to the host. The entries on the free descriptor queues have pre-attached empty buffers whose size and location are described in the original buffer information fields in the descriptor. The host is required to allocate both the descriptor and buffer and pre-link them prior to adding (submitting) a descriptor to one of the available free descriptor queue. The first 32 queues (queue 0 up to queue 31) are reserved for all 30 USB0/1 (endpoints 1 to 15 of each USB module) receive endpoints/channels to handle incoming packets to the device.

15.3.9.3.3 Transmit Submit Queue

Transmit ports use packet queues, referred to as transmit submit queues, to store the packets that are waiting to be transmitted. Each transmit endpoint has dedicated queues (2 queues per port) that are reserved exclusively for a use by a single endpoint. Multiple queues per port/channel are allocated to facilitate quality of service (QoS) for applications that require QoS. The first 30 queues, queue 32 up to queue 61, are allocated for USB0 transmit endpoints 1 to 15 with two queues per endpoint.queues 62 up to queue 91, are allocated for transmit USB1 endpoints 1 to 15.

15.3.9.3.4 Transmit Completion (Return) Queue

Transmit ports use packet queues, referred to as "transmit completion queues, to return packet descriptors to the host after packets have been transmitted. A singe queue is reserved to be used by a single transmit endpoint. Application s/w needs to insure that the right set of queues is used to return the Descriptors after the completion of a packet transmission based on the endpoint number used. For USB0 transmit endpoints 1 to 15 Queues 92 up to queue 106 are reserved/assigned for use as a completion queue respectively. Similarly for USB1 transmit endpoints 1 to 15, queues 107 up to queue 121 is used as completion queue, respectively.

Transmit Completion Queues are also used to return packet Descriptors when performing a Transmit channel teardown operation.

15.3.9.3.5 Receive Completion (Return) Queue

Receive ports use packet queues, referred to as receive completion queues, to return packet descriptors to the port after packets have been received. A singe queue is reserved to be used by a single receive endpoint. Application s/w needs to insure that the right set of queues is used to return the descriptors after the completion of a packet reception based on the endpoint number used. For USB0 receive endpoints 1 to 15 queues 122 up to queue 136 are reserved/assigned for use as a receive completion queue respectively. Similarly for USB1 receive endpoints 1 to 15, queues 137 up to queue 151 is used as completion queue respectively.

Receive channel teardown is not necessary for receive transactions and no channel teardown information nor resource is available

15.3.9.3.6 Diverting Queue Packets from one Queue to Another

The host can move the entire contents of one queue to another queue by writing the source queue number and the destination queue number to the Queue Diversion register. When diverting packets, the descriptors are pushed onto the tail of the destination queue.

15.3.9.3.7 Teardown Queue

Teardown queue

15.3.9.4 Memory Regions and Linking RAM

In addition to allocating buffer for raw data and memory for Descriptors, the host is responsible for allocating additional memory for exclusive use of the CPPI DMA Queue Manager to be used as a scratch PAD RAM. The queue manager uses this memory to manage states of descriptors submitted within the submit queues. In other words, this memory needs not to be managed by the user software and firmware

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

responsibility lies only for allocation/reserving a chunk of memory for the use of the queue manager. The allocated memory can be a single block of memory that is contiguous or two blocks of memory that are not contiguous. These two blocks of memory are referred to as a Linking RAM Regions and should not be confused with memory regions that are used to store descriptors. That is, the use of the term *region* should be used in the context of its use.

To accomplish the linking of submitted descriptors, the queue manager will first resolve the 32-bit descriptor pointer into a 16-bit index which is used for linking and queue pointer purposes. Once the physical index information is determined, the queue manager will link that descriptor onto the descriptor chain that is maintained for that logical queue by writing the linking information out to a linking RAM which is external to the queue manager. The queue manager will also update the queue tail pointer appropriately. Since logical queues within the queue manager are maintained using linked lists, queues cannot become full and no check for fullness is required before queuing a packet descriptor.

The actual physical size of the Linking RAM region(s) to be allocated depends on the total number of descriptors defined within all memory regions. A minimum of four bytes of memory needs to be allocated for each Descriptor defined within all 16 memory regions.

The queue manager has the capability of managing up to 16 memory regions. These memory regions are used to store descriptors of variable sizes. The total number of descriptors that can be managed by the queue manager should not exceed 64K. Each memory region has descriptors of one configurable size, that is, descriptors with different sizes cannot be housed within a single memory region. These 64K descriptors are referenced internally in the queue manager by a 16-bit quantity index.

The information about the Linking RAM regions and the size that are allocated is communicated to the CPPI DMA via three registers dedicated for this purpose. Two of the three registers are used to store the 32-bit aligned start addresses of the Linking RAM regions. The remaining one register is used to store the size of the first Linking RAM. The linking RAM size value stored here is the number of descriptors that is to be managed by the queue manager within that region not the physical size of the buffer, which is four times the number of descriptors.

Note that application is not required to use both linking RAM regions, if the size of the Linking RAM for the first region is large enough to accommodate all descriptors defined. No linking RAM size register for linking RAM region 2 exists. The size of the second linking RAM, when used, is indirectly computed from the total number of descriptors defined less the number of descriptors managed by the first linking RAM

Figure 15-16 displays the relationship between several memory regions and linking RAM.

Figure 15-16. Relationship Between Memory Regions and Linking RAM

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.3.9.5 Zero Length Packets

A special case is the handling of null packets with the CPPI 4.1 compliant DMA controller. Upon receiving a zero length USB packet, the XFER DMA will send a data block to the DMA controller with byte count of zero and the zero byte packet bit of INFO Word 2 set. The DMA controller will then perform normal End of Packet termination of the packet, without transferring data.

If a zero-length USB packet is received, the XDMA will send the CDMA a data block with a byte count of 0 and this bit set. The CDMA will then perform normal EOP termination of the packet without transferring data. For transmit, if a packet has this bit set, the XDMA will ignore the CPPI packet size and send a zero-length packet to the USB controller.

15.3.9.6 CPPI DMA Scheduler

The CPPI DMA scheduler is responsible for controlling the rate and order between the different Tx and Rx threads/transfers that are provided in the CPPI DMA controller. The scheduler table RAM exists within the scheduler.

The DMA controller maintains state information for each of the channels which allows packet segmentation and reassembly operations to be time division multiplexed between channels in order to share the underlying DMA hardware. A DMA scheduler is used to control the ordering and rate at which this multiplexing occurs.

15.3.9.6.1 CPPI DMA Scheduler Operation

Once the scheduler is enabled it will begin processing the entries in the table. When appropriate the scheduler will pass credits to the DMA controller to perform a Tx or Rx operation. The operation of the DMA controller is as follows:

1. After the DMA scheduler is enabled it begins with the table index set to 0.
2. The scheduler reads the entry pointed to by the index and checks to see if the channel in question is currently in a state where a DMA operation can be accepted.
   (a) The DMA channel must be enabled AND
   (b) The CPPI FIFO that the channel talks to has free space on TX (FIFO full signal is not asserted) or a valid block on Rx (FIFO empty signal is not asserted)
3. If the DMA channel is capable of processing a credit to transfer a block, the DMA scheduler will issue that credit via the DMA scheduling interface which is a point to point connection between the DMA Scheduler and the DMA Controller.
   (a) The DMA controller may not be ready to accept the credit immediately and is provided a sched_ready signal which is used to stall the scheduler until it can accept the credit. The DMA controller only asserts the sched_ready signal when it is in the IDLE state.
   (b) Once a credit has been accepted (indicated by sched_req and sched_ready both asserted), the scheduler will increment the index to the next entry and will start again at step 2.
4. If the channel in question is not currently capable of processing a credit, the scheduler will increment the index in the scheduler table to the next entry and will start at step 2.
5. Note that when the scheduler attempts to increment its index, to the value programmed in the table size register, the index will be reset to 0.

15.3.9.6.1.1 CPPI DMA Scheduler Initialization

Before the scheduler can be used, the host is required to initialize and enable the block. This initialization is performed as follows:

1. The Host initializes entries within an internal memory array in the scheduler. This array contains up to 256 entries (4 entries per table word n where n=0-63) and each entry consists of a DMA channel number and a bit indicating if this is a Tx or Rx opportunity. These entries represent both the order and frequency that various Tx and Rx channels will be processed. A table size of 256 entries allows channel bandwidth to be allocated with a maximum precision of 1/256th of the total DMA bandwidth. The more entries that are present for a given channel, the bigger the slice of the bandwidth that channel will be given. Larger tables can be accommodated to allow for more precision. This array can

Copyright © 2011, Texas Instruments Incorporated

only be written by the Host, it cannot be read.

2. If the application does not need to use the entire 256 entries, firmware can initialize the portion of the 256 entries and indicate the size of the entries used by writing onto an internal register in the scheduler which sets the actual size of the array (it can be less than 256 entries).
3. The host writes an internal register bit to enable the scheduler. The scheduler is not required to be disabled in order to change the scheduler array contents.

15.3.9.6.1.2 CPPI DMA Scheduler Programming Examples

Consider a three endpoints use on a system with the following configurations: EP1-Tx, EP2-Rx, and EP2-Tx. Two assumptions are considered:

Case 1: Assume that you would like to service each enabled endpoints (EP1-Tx, EP2-Rx, and EP2-Tx) with equal priority.

The scheduler handles the rate at which an endpoint is serviced by the number of credits programmed (entries) for that particular endpoint within the scheduler Table Words. The scheduler has up to 256 credits that it can grant and for this example the user can configure the number of entries/credits to be anywhere from 3 to 256 with a set of 3 entries.

However, the optimum and direct programming for this scenario would be programming only the first three entries of the scheduler via scheduler Table WORD[0]. Since this case expects the Scheduler to use only the first three entries, you communicate that by programming DMA_SCHED_CTRL.LAST_ENTRY with 2 (that is, 3 -1). The Enabled Endpoint numbers and the data transfer direction is then communicated by programming the first three entries of WORD[0] (ENTRY0_CHANNEL = 1: ENTRY0_RXTX = 0; ENTRY1_CHANNEL = 2: ENTRY1_RXTX = 1;ENTRY2_CHANNEL = 2: ENTRY2_RXTX = 0). With this programming, the Scheduler will only service the first three entries in a round-robin fashion, checking each credited endpoint for transfer one after the other, and servicing the endpoint that has data to transfer.

Case 2: Enabled endpoint EP1-Tx is serviced at twice the rate as the other enabled endpoints (EP2-Rx and EP2-Tx).

The number of entries/credit that has to be awarded to EP1-Tx has to be twice as much of the others. Four entries/credits would suffix to satisfy our requirement with two credits for EP1-Tx, one credit for EP2-Rx, and one credit for EP2-Tx. This requirement is satisfied by allocating any two of the four entries to EP1-Tx endpoint. Again for this example, scheduler Table WORD[0] would suffice since it can handle the first 4 entries. Even though several scenarios exist to programming the order of service for this case, one scenario would be to allow servicing EP1-Tx to back-to-back followed by the other enabled endpoints. Program DMA_SCHED_CTRL.LAST_ENTRY with 3 (that is, 4 -1). Program WORD[0] (ENTRY0_CHANNEL = 1: ENTRY0_RXTX = 0; ENTRY1_CHANNEL = 1: ENTRY1_RXTX = 0; ENTRY2_CHANNEL = 2: ENTRY2_RXTX = 1; ENTRY3_CHANNEL = 2: ENTRY3_RXTX = 0).

15.3.9.7 CPPI DMA State Registers

The port must store and maintain state information for each transmit and receive port/channel. The state information is referred to as the Tx DMA State and Rx DMA State.

15.3.9.7.1 Transmit DMA State Registers

The Tx DMA State is a combination of control fields and protocol specific port scratchpad space used to manipulate data structures and transmit packets. Each transmit channel has two queues. Each queue has a one head descriptor pointer and one completion pointer. There are thirty Tx DMA State registers; one for each port/channel.

The following information is stored in the Tx DMA State:

- Tx Queue Head Descriptor Pointer(s)
- Tx Completion Pointer(s)
- Protocol specific control/status (port scratchpad)

Copyright © 2011, Texas Instruments Incorporated

15.3.9.7.2 Receive DMA State Registers

The Rx DMA State is a combination of control fields and protocol specific port scratchpad space used to manipulate data structures in order to receive packets. Each receive channel has only one queue. Each channel queue has one head descriptor pointer and one completion pointer. There are thirty Rx DMA State registers; one for each port/channel.

The following information is stored in the Rx DMA State:

- Rx Queue Head Descriptor Pointer
- Rx Queue Completion Pointer
- Rx Buffer Offset

15.3.9.8 CPPI DMA Protocols Supported

Four different type of DMA transfers are supported by the CPPI 4.1 DMA; Transparent, RNDIS, Generic RNDIS, and Linux CDC. The following sections will outline the details on these DMA transfer types.

15.3.9.8.1 Transparent DMA Transfer

Transparent Mode DMA operation is the default DMA mode where DMA interrupt is generated whenever a DMA packet is transferred. In the transparent mode, DMA packet size cannot be greater than USB MaxPktSize for the endpoint. This transfer type is ideal for transfer (not packet) sizes that are less than a max packet size.

15.3.9.8.1.1 Transparent DMA Transfer Setup

The following will configure all 30 ports/channels for Transparent DMA Transfer type.

Make sure that RNDIS Mode is disabled globally. This allows the application to configure the CPPI DMA protocol in use to be configured per endpoint need. To disable RNDIS operation, clear RNDIS bit in the USB Control Registers corresponding to the USB Module (default setting), i.e., CTRL0[RNDIS]=0 and CTRL1[RNDIS]=0.

Configure the USB0/1 Tx/Rx DMA Mode Registers (USB0/1 TX(RX)MODE0/1) for the Endpoint field in use is programmed for Transparent Mode, i.e., TXMODE0/1[TXn_MODE] = 00b and RXMODE0/1[RXn_MODE] = 00b.

15.3.9.8.2 RNDIS DMA Transfer

RNDIS mode DMA is used for large transfers (i.e., total data size to be transferred is greater than USB MaxPktSize where the MzxPktSize is a multiple of 64 bytes) that requires multiple USB packets. This is accomplished by breaking the larger packet into smaller packets, where each packet size being USB MaxPktSize except the last packet where its size is less than USB MaxPktSize, including zero bytes. This implies that multiple USB packets of MaxPktSize will be received and transferred together as a single large DMA transfer and the DMA interrupt is generated only at the end of the complete reception of DMA transfer. The protocol defines the end of the complete transfer by receiving a short USB packet (smaller than USB MaxPktSize as mentioned in USB specification 2.0). If the DMA packet size is an exact multiple of USB MaxPktSize, the DMA controller waits for a zero byte packet at the end of complete transfer to signify the completion of the transfer.

NOTE: RNDIS Mode DMA is supported only when USB MaxPktSize is an integral multiple of 64 bytes.

15.3.9.8.2.1 RNDIS DMA Transfer Setup

If all endpoints in use are desired to operate in RNDIS mode, it is only suffix to configure RNDIS DMA operation in RNDIS mode at the global level and application can ignore individual endpoint DMA mode configuration. This is achieved by programming CTRLR0/1[RNDIS] with a ‘1.’

However if DMA mode is required to be configured at the endpoint level, it is required to disable the use of RNDIS at the global level, this is achieved by clearing RNDIS bit field (CTRLR0/1[RNDIS]=0), since global configuration over-rides endpoint configuration.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

To configure RNDIS DMA mode use, configure the field that correspond to the USB module endpoint using the corresponding USB0/1 TX(RX) Mode Register, i.e., TXMODE0/1[TXn_MODE] = 01b and RXMODE0/1[RXn_MODE] = 01b

15.3.9.8.3 Generic RNDIS DMA Transfer

Generic RNDIS DMA transfer mode is identical to the normal RNDIS mode in nearly all respects, except for the exception case where the last packet of the transfer can either be a short packet or the MaxPktSize. When the last packet size is equal to the MaxPktSize then no additional zero-byte packet is sent when using Generic RNDIS transfer. Generic RNDIS transfer makes use of a USB0/1 GENERIC RNDIS EPn Size register (there exists a register for each endpoint) that must be programmed with the transfer size (in bytes) of the transfer for the USB Module (USB0 or USB1) prior to posting a transfer transaction. If the transfer size is an integer multiple of USB MaxPktSize then no additional zero-byte packet is sent when using Generic RNDIS transfer. However, if the a short packet has been sent prior to programmed size count, the transfer would end in a similar fashion an RDIS transfer would behave. For example, it the USB MaxPktSize (Tx/RxMaxP) is programmed with a value of 64, the Generic RNDIS EP Size register for that endpoint must be programmed with a value that is an integer multiple of 64 (for example, 64, 128, 192, 256, etc.) for it behave differently than RNDIS transfer. In other words, when using Generic RNDIS mode and the DMA is tasked to transfer data transfer size that is less or equal the size value programmed within the USB0/1 GENERIC RNDIS EPn Size register.

This means that Generic RNDIS mode will perform data transfer in the same manner as RNDIS mode, closing the CPPI packet if a USB packet with a size less than the USB MaxPktSize size value is received. Otherwise, the packet will be closed when the value in the Generic RNDIS EP Size register is reached.

Using USB0/1 GENERIC RNDIS EPn Size register, a packet of up to 64K bytes (65536 bytes) can be transferred. This is to allow the host software to program the USB module to transfer data that is an exact multiple of the USB MaxPktSize (Tx/RxMaxP programmed value) without having to send an additional short packet to terminate.

NOTE: As in RNDIS mode, the USB max packet size (Tx/RxMaxp programmed value) of any Generic RNDIS mode enabled endpoints must be a multiple of 64 bytes. Generic RNDIS acceleration should not be enabled for endpoints where the max packet size is not a multiple of 64 bytes. Only transparent mode should be used for such endpoints.

15.3.9.8.3.1 Generic RNDIS DMA Transfer Setup

Disable the use of RNDIS at the global level, this is achieved by clearing RNDIS bit field (CTRLR0/1[RNDIS]=0), since global configuration over-rides endpoint configuration.

Configure the field that correspond to the USB module endpoint using the corresponding USB0/1 TX(RX) Mode Register, i.e., TXMODE0/1[TXn_MODE] = 11b and RXMODE0/1[RXn_MODE] = 11b.

15.3.9.8.4 Linux CDC DMA Transfer

Linux CDC DMA transfer mode acts in the same manner as RNDIS packets, except for the case where the last data matches the max USB packet size requiring additional zero-byte packet transfer in RNDIS mode and this is not the case for Linux CDC. If the last data packet of a transfer is a short packet where the data size is greater than zero and less the USB MaxPktSize, then the behavior of the Linux CDC DMA transfer type is identical with the RNDIS DMA transfer type. The only exception is when the short packet length terminating the transfer is a Null Packet. In this case, instead of transferring the Null Packet, it will transfer a data packet of size 1 byte with the data value of 00h.

In transmit operation, if an endpoint is configured or CDC Linux mode, upon receiving a Null Packet from the CPPI DMA, the XFER DMA will then generate a packet containing 1 byte of data, whose value is 00h, indicating the end of the transfer. During receive operation, the XFER DMA will recognize the one byte zero packet as a termination of the data transfer, and sends a block of data with the EOP indicator set and a byte count of one to the CPPI DMA controller. The CPPI DMA realizing the end of the transfer termination will not update/increase the packet size count of the Host Packet Descriptor.

Copyright © 2011, Texas Instruments Incorporated

15.3.9.8.4.1 Linux CDC DMA Transfer Setup

Disable the use of RNDIS at the global level, this is achieved by clearing RNDIS bit field (CTRLR0/1[RNDIS]=0), since global configuration over-rides endpoint configuration.

Configure the field that correspond to the USB module endpoint using the corresponding USB0/1 TX(RX) Mode Register, i.e., TXMODE0/1[TXn_MODE] = 10b and RXMODE0/1[RXn_MODE] = 10b.

15.3.9.9 USB Data Flow Using DMA

The necessary steps required to perform a USB data transfer using the CPPI 4.1 DMA is expressed using an example for both transmit and receive cases. Assume USB0 is ready to perform a USB data transfer of size 608 bytes (see Figure 15-17).

The 608 bytes data to be transferred is described using three descriptors. Since each data buffer defined within a single descriptor has a size of 256 bytes, a 608 bytes data buffer would require three descriptors.

Figure 15-17. High-level Transmit and Receive Data Transfer Example

Example assumptions:

- The CPPI data buffers are 256 bytes in length.
- USB0 module is to be used to perform this transfer. Note that the steps required is similar for USB1 use.
- The USB0 endpoint 1 Tx and Rx endpoint 1 size are programmed to handle max USB packet size of 512 bytes.
- A single transfer length is 608 bytes.
- The SOP offset is 0.

The above translates to the following multi-descriptor setup:

Transmit Case

Transmit setup is as follows:

- One packet descriptor with packet length (this is NOT data buffer length, the term packet used here is to mean a transfer length not USB packet) field of 608 bytes and a Data Buffer of size 256 Bytes linked to the 1st host buffer descriptor.
- Two buffer descriptors with first buffer descriptor (this is the one linked to the packet descriptor) defining the second data buffer size of 256 Bytes which in turn is linked to the next (second) buffer descriptor.

Copyright © 2011, Texas Instruments Incorporated

- Second buffer fescriptor with a fata buffer size of 96 bytes (can be greater, the packet descriptor contain the size of the packet) linked to no other descriptor (NULL).

Receive Case

For this example since each data buffer size is 256 bytes, we would require a minimum of three descriptors that would define data buffer size of 608 bytes. The receive setup is as follows:

- Two buffer descriptors with 256 bytes of data buffer size
- One buffer descriptor with 96 bytes (can be greater) of data buffer size

Within the rest of this section, the following nomenclature is used:

BD — Buffer Descriptor or Host Buffer Descriptor

DB — Data Buffer Size of 256 Bytes

PBD — Pointer to Host Buffer Descriptor

PD — Host Packet Descriptor

PPD — Pointer to Host Packet Descriptor

TXSQ — Transmit Queue or Transmit Submit Queue (for USB0 EP1, use Queues 32 or 33, for USB1 EP1 use Queues 62 or 63)

TXCQ — Transmit Completion Queue or Transmit Return Queue (for USB0 Tx EP1, use Queue 93, and for USB1 Tx EP1 use Queue 125)

RXCQ — Receive Completion Queue or Receive Return Queue (for USB0 Rx EP1, use Queue 109, for USB1 Rx EP1 use Queue 141)

RXSQ — Receive Free/Buffer Descriptor Queue or Receive Submit Queue. (For USB0 Rx EP1 Queue 0 is used and for USB1 Rx EP1 Queue 16 should be used

15.3.9.9.1 Transmit USB Data Flow Using DMA

The transmit descriptors and queue status configuration prior to the transfer taking place is shown in Figure 15-18.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 15-18. Transmit Descriptors and Queue Status Configuration

15.3.9.9.1.1 Transmit Initialization (Step 1)

The CPU performs the following steps for transmit initialization:

1. Initializes Memory Region 0 base address and Memory Region 0 size, Link RAM0 Base address, Link RAM0 data size, and Link RAM1 Base address.
2. Creates PD, BDs, and DBs in main memory and link as indicated in Figure 15-19.
3. Initializes and configures the Queue Manager, Channel Setup, DMA Scheduler, and Mentor USB 2.0 Core.
4. Adds (pushes) the PPD and the two PBDs to the TXSQ by writing the Packet Descriptor address to the TXSQ CTRL D Register.

Figure 15-19 captures the BD/DB pair in main memory and later submitted within the TXSQ.

Copyright © 2011, Texas Instruments Incorporated

Figure 15-19. Transmit USB Data Flow Example (Initialization)

15.3.9.9.1.2 CDMA and XDMA Transfer Packet Data Into Endpoint FIFO (Step 2)

The steps for CDMA and XDMA to transfer packet data into endpoint FIFO are as follows:

1. The Queue Manager informs the CDMAS that the TXSQ is not empty.
2. CDMAS checks that the CPPI FIFO FIFO_full is not asserted, then issues a credit to the CDMA.
3. CDMA reads the Packet Descriptor pointer (PPD) and descriptor size hint from the queue manager
4. For each 64-byte block of data in the packet data payload (note that packet refers here to CPPI packet which is not the same as USB packet and it means to refer to data transfer size):
   (a) The CDMA transfers a max burst of 64-byte block (OCP burst) from the data to be transferred in main memory to the CPPI FIFO
   (b) The XDMA sees FIFO_empty not asserted and transfers 64-byte block from CPPI FIFO to Endpoint FIFO.
   (c) The CDMA performs the above two steps (‘a’ and ‘b’) three more times since the data size of the HPD is 256 bytes.
5. The CDMA reads the first buffer descriptor pointer (PBD).
6. For each 64-byte block of data in the packet data payload:
   (a) The CDMA transfers a max burst of 64-byte block from the data to be transferred in main memory to the CPPI FIFO.
   (b) The XDMA sees FIFO_empty not asserted and transfers 64-byte block from CPPI FIFO to Endpoint FIFO.
   (c) The CDMA performs the above two steps (‘a’ and ‘b’) two more times since data size of the HBD is 256 bytes.
7. The CDMA reads the second Buffer Descriptor pointer (PBD
8. For each 64-byte block of data in the packet data payload:
   (a) The CDMA transfers a max burst of 64-byte block from the data to be transferred in main memory to the CPPI FIFO.
   (b) The XDMA sees FIFO_empty not asserted and transfers 64-byte block from CPPI FIFO to Endpoint FIFO.
   (c) The CDMA transfers the last remaining 32-byte from the data to be transferred in main memory to the CPPI FIFO.
   (d) The XDMA sees FIFO_empty not asserted and transfers 32-byte block from CPPI FIFO to Endpoint FIFO.

Copyright © 2011, Texas Instruments Incorporated

15.3.9.9.1.3 USB 2.0 Core Transmits USB Packets for Tx (Step 3)

1. Once the XDMA has transferred enough 64-byte blocks of burst of data from the CPPI FIFO to fill the Endpoint FIFO (accumulated 512 bytes), it signals the Mentor USB 2.0 Core that a TX packet is ready (sets the endpoint's TxPktRdy bit).
2. The USB 2.0 Core will transmit the packet from the Endpoint FIFO out on the USB BUS when it receives a corresponding IN request from the attached USB Host.
3. After the USB packet is transferred, the USB 2.0 Core issues a TX DMA_req to the XDMA.
4. This process is repeated until (the above steps '1', '2', and '3') the entire packet has been transmitted. The XDMA will also generate the required termination packet depending on the termination mode configured for the endpoint.

15.3.9.9.1.4 Return Packet to Completion Queue and Interrupt CPU for Tx (Step 4)

1. After all data for the packet has been transmitted (as specified by the packet size field), the CDMA will write the pointer of the Packet Descriptor only to the TX Completion Queue specified in the return queue manager / queue number fields of the packet descriptor.
2. The Queue Manager then indicates the status of the TXSQ (empty) to the CDMAS and the TXCQ to the CPU via an interrupt.

15.3.9.9.2 Receive USB Data Flow Using DMA

The receive buffer descriptors and queue status configuration prior to the transfer taking place is shown in Figure 15-20.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 15-20. Receive Buffer Descriptors and Queue Status Configuration

15.3.9.9.2.1 Receive Initialization (Step 1)

1. The CPU initializes Queue Manager with the Memory Region 0 base address and Memory Region 0 size, Link RAM0 Base address, Link RAM0 data size, and Link RAM1 Base address.
2. The CPU creates BDs, and DBs in main memory and link them creating a BD and DB pairs.
3. CPU then initializes the RXCQ queue and configures the Queue Manager, Channel Setup, DMA Scheduler, and USB 2.0 Core.
4. It then adds (pushes) the address of the addresses/pointers of three Buffer Descriptors (PBDs) into the

Copyright © 2011, Texas Instruments Incorporated

RXSQ by writing onto CTRL D register.

Figure 15-21 displays receive operation Initialization.

Figure 15-21. Receive USB Data Flow Example (Initialization)

15.3.9.9.2.2 USB 2.0 Core Receives a Packet, XDMA Starts Data Transfer for Receive (Step 2)

1. The USB 2.0 Core receives a USB packet from the USB Host and stores it in the Endpoint FIFO.
2. It then asserts a DMA_req to the XDMA informing it that data is available in the Endpoint FIFO.
3. The XDMA verifies the corresponding CPPI FIFO is not full via the FIFO_full signal, then starts transferring 64-byte data blocks (burst) from the Endpoint FIFO into the CPPI FIFO.

15.3.9.9.2.3 CDMA Transfers Data from SSRAM / PPU to Main Memory for Receive (Step 3)

1. The CDMAS see FIFO_empty de-asserted (there is RX data in the FIFO) and issues a transaction credit to the CDMA.
2. The CDMA begins packet reception by fetching the first PBD from the Queue Manager using the Free Descriptor / Buffer Queue 0 (Rx Submit Queue) index that was initialized in the RX port DMA state for that channel.
3. The CDMA will then begin writing the 64-byte block of packet data into this DB.
4. The CDMA will continue filling the buffer with additional 64-byte blocks of data from the CPPI FIFO and will fetch additional PBD as needed using the Free Descriptor / Buffer Queue 1, 2, and 3 indexes for the 2nd, 3rd, and remaining buffers in the packet. After each buffer is filled, the CDMA writes the buffer descriptor to main memory.

15.3.9.9.2.4 CDMA Completes the Packet Transfer for Receive (Step 4)

1. After the entire packet has been received, the CDMA writes the packet descriptor to main memory.
2. The CDMA then writes the packet descriptor to the RXCQ specified in the Queue Manager / Queue Number fields in the RX Global Configuration Register.
3. The Queue Manager then indicates the status of the RXCQ to the CPU via an interrupt.
4. The CPU can then process the received packet by popping the received packet information from the RXCQ and accessing the packet's data from main memory.

Copyright © 2011, Texas Instruments Incorporated

15.3.10 USB 2.0 Test Modes

The USB2.0 controller supports the four USB 2.0 test modes (Test_SE0_NAK, Test_J, Test_K, and Test_Packet) defined for high-speed functions. It also supports an additional “FIFO access” test mode that can be used to test the operation of the CPU interface, the DMA controller and the RAM block.

The test modes are entered by writing to the TESTMODE register. A test mode is usually requested by the host sending a SET_FEATURE request to Endpoint 0. When the software receives the request, it should wait until the Endpoint 0 transfer has completed (when it receives the Endpoint 0 interrupt indicating the status phase has completed) then write to the TESTMODE register.

Note: These test modes have no purpose in normal operation.

15.3.10.1 TEST_SE0_NAK

To enter the Test_SE0_NAK test mode, the software should set the TEST_SE0_NAK bit in the TESTMODE register to 1. The USB controller will then go into a mode in which it responds to any valid IN token with a NAK.

15.3.10.2 TEST_J

To enter the Test_J test mode, the software should set the TEST_J bit in the TESTMODE register to 1. The USB controller will then go into a mode in which it transmits a continuous J on the bus.

15.3.10.3 TEST_K

To enter the Test_K test mode, the software should set the TEST_K bit in the TESTMODE register to 1. The USB controller will then go into a mode in which it transmits a continuous K on the bus.

15.3.10.4 TEST_PACKET

To execute the Test_Packet, the software should:

1. Start a session (if the core is being used in Host mode.
2. Write the standard test packet (shown below) to the Endpoint 0 FIFO.
3. Write 8h to the TESTMODE register (TEST_PACKET = 1) to enter Test_Packet test mode.
4. Set the TxPktRdy bit in the HOST/PERI_CSR0 register (bit 1).

The 53 byte test packet to load is as follows (all bytes in hex). The test packet only has to be loaded once; the USB controller will keep re-sending the test packet without any further intervention from the software.

Table 15-28 displays the 53 Bytes test packet content.

Table 15-28. 53 Bytes Test Packet Content

0	0	0	0	0	0	0	0
0	AA	AA	AA	AA	AA	AA	AA
AA	EE	EE	EE	EE	EE	EE	EE
EE	FE	FF	FF	FF	FF	FF	FF
FF	FF	FF	FF	FF	7F	BF	DF
EF	F7	FB	FD	FC	7E	BF	DF
EF	F7	FB	FD	7E			

This data sequence is defined in Universal Serial Bus Specification Revision 2.0, Section 7.1.20. The USB controller will add the DATAA0 PID to the head of the data sequence and the CRC to the end.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.3.10.5 FIFO_ACCESS

The FIFO Access test mode allows you to test the operation of CPU Interface, the DMA controller (if configured), and the RAM block by loading a packet of up to 64 bytes into the Endpoint 0 FIFO and then reading it back out again. Endpoint 0 is used because it is a bidirectional endpoint that uses the same area of RAM for its Tx and Rx FIFOs.

NOTE: The core does not need to be connected to the USB bus to run this test. If it is connected, then no session should be in progress when the test is run.

The test procedure is as follows:

1. Load a packet of up to 64 bytes into the Endpoint 0 Tx FIFO.
2. Set HOST/PERI_CSR0[TXPKTRDY].
3. Write 40h to the TESTMODE register (FIFO_ACCESS = 1).
4. Unload the packet from the Endpoint Rx FIFO, again.
5. Set HOST/PERI_CSR0[SERVICEDRXPKTRDY].

Writing 40h to the TESTMODE register causes the following sequence of events:

The Endpoint 0 CPU pointer (that records the number of bytes to be transmitted) is copied to the Endpoint 0 USB pointer (that records the number of bytes received).

1. The Endpoint 0 CPU pointer is reset.
2. HOST/PERI_CSR0[TXPKTRDY] is cleared.
3. HOST/PERI_CSR0[RXPKTRDY] is set.
4. An Endpoint 0 interrupt is generated (if enabled).

The effect of these steps is to make the Endpoint 0 controller act as if the packet loaded into the Tx FIFO has flushed and the same packet received over the USB. The data that was loaded in the Tx FIFO can now be read out of the Rx FIFO.

15.3.10.6 FORCE_HOST

The Force Host test mode enables you to instruct the core to operate in Host mode, regardless of whether it is actually connected to any peripheral; that is, the state of the CID input and the LINESTATE and HOSTDISCON signals are ignored. (While in this mode, the state of the HOSTDISCON signal can be read from the BDEVICE bit in the device control register (DEVCTL)) .

This mode, which is selected by writing 80h to the TESTMODE register (FORCE_HOST = 1), allows implementation of the USB Test_Force_Enable (USB 2.0 Specification Section 7.1.20). It can also be used for debugging PHY problems in hardware.

While the FORCE_HOST bit remains set, the core enters the Host mode when the SESSION bit in DEVCTL is set to 1 and remains in the Host mode until the SESSION bit is cleared to 0 even if a connected device is disconnected during the session. The operating speed while in this mode is determined by the FORCE_HS and FORCE_FS bits in the TESTMODE register.

15.4 Supported Use Cases

The USB Subsystem supports two independent USB Modules where each USB module can operate as a host or peripheral. When operating as a host, it is capable of interfacing to a single target/peripheral directly or to multiple targets via hub at low-, full-, or high-speed. When operating as a peripheral, it is capable of interfacing to a host at a full- or high-speed.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5 USB Registers

15.5.1 USBSS Registers

The USBSS registers contain the registers that are used to control at the global level and apples to all submodules. Table 15-29 lists the registers for the USBSS submodule.

Table 15-29. USBSS Registers

Address Offset	Acronym	USBSS Register
0h	REVREG	USBSS REVISION
10h	SYSCONFIG	USBSS SYSCONFIG
20h	EOI	USBSS IRQ_EOI
24h	IRQSTATRAW	USBSS IRQ_STATUS_RAW
28h	IRQSTAT	USBSS IRQ_STATUS
2Ch	IRQENABLER	USBSS IRQ_ENABLE_SET
30h	IRQCLEARR	USBSS IRQ_ENABLE_CLR
100h	IRQDMATHOLDTX00	USBSS IRQ_DMA_THRESHOLD_TX0_0
104h	IRQDMATHOLDTX01	USBSS IRQ_DMA_THRESHOLD_TX0_1
108h	IRQDMATHOLDTX02	USBSS IRQ_DMA_THRESHOLD_TX0_2
10Ch	IRQDMATHOLDTX03	USBSS IRQ_DMA_THRESHOLD_TX0_3
110h	IRQDMATHOLDRX00	USBSS IRQ_DMA_THRESHOLD_RX0_0
114h	IRQDMATHOLDRX01	USBSS IRQ_DMA_THRESHOLD_RX0_1
118h	IRQDMATHOLDRX02	USBSS IRQ_DMA_THRESHOLD_RX0_2
11Ch	IRQDMATHOLDRX03	USBSS IRQ_DMA_THRESHOLD_RX0_3
120h	IRQDMATHOLDTX10	USBSS IRQ_DMA_THRESHOLD_TX1_0
124h	IRQDMATHOLDTX11	USBSS IRQ_DMA_THRESHOLD_TX1_1
128h	IRQDMATHOLDTX12	USBSS IRQ_DMA_THRESHOLD_TX1_2
12Ch	IRQDMATHOLDTX13	USBSS IRQ_DMA_THRESHOLD_TX1_3
130h	IRQDMATHOLDRX10	USBSS IRQ_DMA_THRESHOLD_RX1_0
134h	IRQDMATHOLDRX11	USBSS IRQ_DMA_THRESHOLD_RX1_1
138h	IRQDMATHOLDRX12	USBSS IRQ_DMA_THRESHOLD_RX1_2
13Ch	IRQDMATHOLDRX13	USBSS IRQ_DMA_THRESHOLD_RX1_3
140h	IRQDMAENABLE0	USBSS IRQ_DMA_ENABLE_0
144h	IRQDMAENABLE1	USBSS IRQ_DMA_ENABLE_1
200h	IRQFRAMETHOLDTX00	USBSS IRQ_FRAME_THRESHOLD_TX0_0
204h	IRQFRAMETHOLDTX01	USBSS IRQ_FRAME_THRESHOLD_TX0_1
208h	IRQFRAMETHOLDTX02	USBSS IRQ_FRAME_THRESHOLD_TX0_2
20Ch	IRQFRAMETHOLDTX03	USBSS IRQ_FRAME_THRESHOLD_TX0_3
210h	IRQFRAMETHOLDRX00	USBSS IRQ_FRAME_THRESHOLD_RX0_0
214h	IRQFRAMETHOLDRX01	USBSS IRQ_FRAME_THRESHOLD_RX0_1
218h	IRQFRAMETHOLDRX02	USBSS IRQ_FRAME_THRESHOLD_RX0_2
21Ch	IRQFRAMETHOLDRX03	USBSS IRQ_FRAME_THRESHOLD_RX0_3
220h	IRQFRAMETHOLDTX10	USBSS IRQ_FRAME_THRESHOLD_TX1_0
224h	IRQFRAMETHOLDTX11	USBSS IRQ_FRAME_THRESHOLD_TX1_1
228h	IRQFRAMETHOLDTX12	USBSS IRQ_FRAME_THRESHOLD_TX1_2
22Ch	IRQFRAMETHOLDTX13	USBSS IRQ_FRAME_THRESHOLD_TX1_3
230h	IRQFRAMETHOLDRX10	USBSS IRQ_FRAME_THRESHOLD_RX1_0
234h	IRQFRAMETHOLDRX11	USBSS IRQ_FRAME_THRESHOLD_RX1_1
238h	IRQFRAMETHOLDRX12	USBSS IRQ_FRAME_THRESHOLD_RX1_2
23Ch	IRQFRAMETHOLDRX13	USBSS IRQ_FRAME_THRESHOLD_RX1_3
240h	IRQFRAMEENABLE0	USBSS IRQ_FRAME_ENABLE_0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-29. USBSS Registers (continued)

Address Offset	Acronym	USBSS Register
244h	IRQFRAMEENABLE1	USBSS IRQ_FRAME_ENABLE_1

15.5.1.1 USBSS Revision Register (REVREG)

The revision register contains the major and minor revisions for the USB subsystem module. The USBSS revision register is shown in Figure 15-22 and described in Table 15-30.

Figure 15-22. USBSS Revision Register (REVREG)

31 30	29 28	27 16
Scheme	Reserved	func
R-1	R-0h	R-EA2h

15 11	10 8	7 6	5 0
R_rtl	X_major	Custom	Y_minor
R-1	R-0h	R-0h	R-7h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-30. USBSS Revision Register (REVREG) Field Description

Bits	Field	Value	Description
31-30	Scheme		Used to distinguish between old scheme and current.
		0	Legacy ASP or WTBU
		1	Highlander 0.8
29-28	Reserved	0	Reserved
27-16	func		Function indicates a software compatible module family.
15-11	R_rtl		RTL revision. Will vary depending on release.
10-8	X_major		Major revision.
7-6	Custom		Custom revision
5-0	Y_minor		Minor revision.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.1.2 USBSS SYSCONFIG Register (SYSCONFIG)

The SYSCONFIG register is the clock management configuration register for the USBSS.

The clock enable bits 8 to 11 should only be configured/modified during the firmware initialization stage of the USBSS. In other words, these bits should not be modified while the system is running or after enabling the clock. These bits control the clock generation modules. Dynamically setting these bits can disable the power management logic. The USBSS SYSCONFIG register is shown in shown in Figure 15-23 and described in Table 15-31.

Figure 15-23. USBSS SYSCONFIG Register (SYSCONFIG)

31							16
Reserved							
R-0h							

15			12	11	10	9	8
Reserved				usb0 ocp en_n	phy0 utmi en_n	usb1 ocp en_n	phy1 utmi en_n
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
Reserved		Standby mode		Idlemode		Freeemu	Soft reset
R-0h		R/W-2h		R/W-2h		R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-31. USBSS SYSCONFIG Register (SYSCONFIG) Field Descriptions

Bits	Field	Value	Description
31-12	Reserved		Reserved
11	usb0 ocp en_n		Active low clock enable for usb0_ocp_clk
		0	Enable
		1	Disable
10	phy0 utmi en_n		Active low clock enable for phy0_utmi_clk
		0	Enable
		1	Disable
9	usb1 ocp en_n		Active low clock enable for usb1_ocp_clk
		0	Enable
		1	Disable
8	phy1 utmi en_n		Active low clock enable for phy1_utmi_clk
		0	Enable
		1	Disable
7-6	Reserved		Reserved
5-4	Standby mode		Configuration of the local initiator state management mode.
		0	Force-standby mode
		1	No-standby mode
		2	Smart-standby mode
		3	Smart-standby wakeup capable mode
3-2	Idlemode		Configuration of the local target state management mode.
		0	Force-idle mode
		1	No-idle mode
		2	Smart-idle mode
		3	Smart-idle wakeup capable mode
1	Freeemu		Sensitivity to emulation (debug) suspend input signal.
		0	Sensitive to emulation suspend
		1	NOT sensitive to emulation suspend

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-31. USBSS SYSCONFIG Register (SYSCONFIG) Field Descriptions (continued)

Bits	Field	Value	Description
0	Soft reset		Software reset of USBSS, USB0, and USB1 modules
		Write 0	No action
		Write 1	Initiate software reset
		Read 0	Reset done, no action
		Read 1	Reset ongoing

15.5.1.3 USBSS End of Interrupt Register (EOI)

The USBSS end of interrupt register (EOI) allows the CPU to acknowledge completion of an interrupt by writing 0, zero, to the EOI_VECTOR field. An eoi_write signal will be generated and another interrupt will be triggered if interrupt sources remain.

This register will be reset one cycle after it has been written to. The USBSS end of interrupt register is shown in Figure 15-24 and described in Table 15-32.

Figure 15-24. USBSS End of Interrupt Register (EOI)

31	16
Reserved	
R-0h	

15	1	0
Reserved		EOI_VECTOR
R-0h		R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-32. USBSS End of Interrupt Register (EOI) Field Descriptions

Bits	Field	Description
31-1	Reserved	Always read as 0. Writes have no effect.
0	EOI_VECTOR	EOI for USBSS interrupt

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.1.4 USBSS IRQ_STATUS_RAW (IRQSTATRAW)

The USBSS IRQSTATRAW register allows the USBSS interrupt sources to be manually triggered when writing a 1 to a specific bit. A read from this register returns the USBSS interrupt event pending value. This register is shown in Figure 15-25 and described in Table 15-33.

General actions per bit:
Write 0: No action
Write 1: Set event
Read 0: No event pending
Read 1: Event pending

Figure 15-25. USBSS IRQ_STATUS_RAW (IRQSTATRAW)

31					16
Reserved					
R-0h					

15	12	11	10	9	8
Reserved		rx_pkt_cmp_1	tx_pkt_cmp_1	rx_pkt_cmp_0	tx_pkt_cmp_0
R-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	3	2	1	0
Reserved		pd_cmp_flag	rx_mop_ starvation	rx_sop_ starvation
R-0h		R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-33. USBSS IRQ_STATUS_RAW (IRQSTATRAW) Field Descriptions

Bits	Field	Description
31-12	Reserved	Always read as 0. Writes have no effect.
11	rx_pkt_cmp_1	Interrupt status for USB1 Rx CPPI DMA packet completion status
10	tx_pkt_cmp_1	Interrupt status for USB1 Tx CPPI DMA packet completion status
9	rx_pkt_cmp_0	Interrupt status for USB0 Rx CPPI DMA packet completion status
8	tx_pkt_cmp_0	Interrupt status for USB0 Tx CPPI DMA packet completion status
7-3	Reserved	Always read as 0. Writes have no effect.
2	pd_cmp_flag	Interrupt status when the packet is completed and the Packet Descriptor is pushed into the queue manager
1	rx_mop_starvation	Interrupt status when queue manager cannot allocate an Rx buffer in the middle of a packet.
0	rx_sop_starvation	Interrupt status when queue manager cannot allocate an Rx buffer at the start of a packet.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.1.5 USBSS IRQ_STATUS Register (IRQSTAT)

The USBSS IRQSTAT register contains the interrupt sources status of the enabled interrupt. IRQSTAT register is a subset of IRQSTATRAW register. Writing a '1' to a bit field will clear a pending interrupt event. This register is shown in Figure 15-26 and described in Table 15-34.

General actions per bit:
Write 0: No action
Write 1: Clear event
Read 0: No event pending
Read 1: Event pending

Figure 15-26. USBSS IRQ_STATUS (IRQSTAT)

31					16
Reserved					
R-0h					

15	12	11	10	9	8
Reserved		rx_pkt_cmp_1	tx_pkt_cmp_1	rx_pkt_cmp_0	tx_pkt_cmp_0
R-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	3	2	1	0
Reserved		pd_cmp_flag	rx_mop_ starvation	rx_sop_ starvation
R-0h		R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-34. USBSS IRQ_STATUS (IRQSTAT) Field Descriptions

Bits	Field	Description
31-12	Reserved	Always read as 0. Writes have no effect.
11	rx_pkt_cmp_1	Interrupt status for USB1 Rx CPPI DMA packet completion status
10	tx_pkt_cmp_1	Interrupt status for USB1 Tx CPPI DMA packet completion status
9	rx_pkt_cmp_0	Interrupt status for USB0 Rx CPPI DMA packet completion status
8	tx_pkt_cmp_0	Interrupt status for USB0 Tx CPPI DMA packet completion status
7-3	Reserved	Always read as 0. Writes have no effect.
2	pd_cmp_flag	Interrupt status when the packet is completed and the Packet Descriptor is pushed into the queue manager.
1	rx_mop_starvation	Interrupt status when queue manager cannot allocate an Rx buffer in the middle of a packet.
0	rx_sop_starvation	Interrupt status when queue manager cannot allocate an Rx buffer at the start of a packet.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.1.6 USBSS IRQ_ENABLE_SET Register (IRQENABLER)

The USB interrupt mask set register (IRQENABLER) allows the USB interrupts to be enabled (unmasked). A read from this register returns the USB interrupts enabled.

The USBSS IRQ_ENABLE_SET register is shown in Figure 15-27 and described in Table 15-35.

General actions per bit:
Write 0: No action
Write 1: Interrupt enabled
Read 0: Interrupt disabled
Read 1: Interrupt enabled

Figure 15-27. USBSS IRQ_ENABLE_SET Register (IRQENABLER)

31					16
Reserved					
R-0h					

15	12	11	10	9	8
Reserved		rx_pkt_cmp_1	tx_pkt_cmp_1	rx_pkt_cmp_0	tx_pkt_cmp_0
R-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	3	2	1	0
Reserved		pd_cmp_flag	rx_mop_ starvation	rx_sop_ starvation
R-0h		R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-35. USBSS IRQ_ENABLE_SET Register (IRQENABLER) Field Descriptions

Bits	Field	Description
31-12	Reserved	Always read as 0. Writes have no effect.
11	rx_pkt_cmp_1	Interrupt enable for USB1 Rx CPPI DMA packet completion status
10	tx_pkt_cmp_1	Interrupt enable for USB1 Tx CPPI DMA packet completion status
9	rx_pkt_cmp_0	Interrupt enable for USB0 Rx CPPI DMA packet completion status
8	tx_pkt_cmp_0	Interrupt enable for USB0 Tx CPPI DMA packet completion status
7-3	Reserved	Always read as 0. Writes have no effect.
2	pd_cmp_flag	Interrupt enable when the packet is completed and the Packet Descriptor is pushed into the queue manager.
1	rx_mop_starvation	Interrupt enable when queue manager cannot allocate an Rx buffer in the middle of a packet.
0	rx_sop_starvation	Interrupt enable when queue manager cannot allocate an Rx buffer at the start of a packet.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.1.7 USBSS IRQ_ENABLE_CLR Register (IRQCLEARR)

The USBSS IRQCLEARR register allows the USBSS interrupt sources to be masked when writing a 1 to a specific bit. A read of this register returns the USBSS interrupt enabled value.

The USBSS IRQ_ENABLE_CLR register is shown in Figure 15-28 and described in Table 15-36.

General actions per bit:
Write 0: No action
Write 1: Disable interrupt
Read 0: Interrupt disabled
Read 1: Interrupt enabled

Figure 15-28. USBSS IRQ_ENABLE_CLR Register (IRQCLEARR)

31					16
Reserved					
R-0h					

15	12	11	10	9	8
Reserved		rx_pkt_cmp_1	tx_pkt_cmp_1	rx_pkt_cmp_0	tx_pkt_cmp_0
R-0h		R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	3	2	1	0
Reserved		pd_cmp_flag	rx_mop_ starvation	rx_sop_ starvation
R-0h		R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-36. USBSS IRQ_ENABLE_CLR Register (IRQCLEARR) Field Descriptions

Bits	Field	Description
31-12	Reserved	Always read as 0. Writes have no effect.
11	rx_pkt_cmp_1	Interrupt enable for USB1 Rx CPPI DMA packet completion status
10	tx_pkt_cmp_1	Interrupt enable for USB1 Tx CPPI DMA packet completion status
9	rx_pkt_cmp_0	Interrupt enable for USB0 Rx CPPI DMA packet completion status
8	tx_pkt_cmp_0	Interrupt enable for USB0 Tx CPPI DMA packet completion status
7-3	Reserved	Always read as 0. Writes have no effect.
2	pd_cmp_flag	Interrupt enable when the packet is completed and the Packet Descriptor is pushed into the queue manager.
1	rx_mop_starvation	Interrupt enable when queue manager cannot allocate an Rx buffer in the middle of a packet.
0	rx_sop_starvation	Interrupt enable when queue manager cannot allocate an Rx buffer at the start of a packet.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.1.8 USBSS IRQ_DMA_THRESHOLD_TX0_0 Register (IRQDMATHOLDTX00)

The USBSS IRQDMATHOLDTX00 register defines the size of the four DMA thresholds for interrupt pacing for USB0. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_TX0_0 register is shown in Figure 15-29 and described in Table 15-37.

Figure 15-29. USBSS IRQ_DMA_THRESHOLD_TX0_0 Register (IRQDMATHOLDTX00)

31–24	23–16	15–8	7–0
dma_thres_tx0_3	dma_thres_tx0_2	dma_thres_tx0_1	Reserved
R/W-0h	R/W-0h	R/W-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-37. USBSS IRQ_DMA_THRESHOLD_TX0_0 Register (IRQDMATHOLDTX00) Field Descriptions

Bits	Field	Description
31-24	dma_thres_tx0_3	DMA threshold value for tx_pkt_cmp_0 for USB0 Endpoint 3
23-16	dma_thres_tx0_2	DMA threshold value for tx_pkt_cmp_0 for USB0 Endpoint 2
15-8	dma_thres_tx0_1	DMA threshold value for tx_pkt_cmp_0 for USB0 Endpoint 1
7-0	Reserved	Always read as 0. Writes have no effect.

15.5.1.9 USBSS IRQ_DMA_THRESHOLD_TX0_1 Register (IRQDMATHOLDTX01)

The USBSS IRQDMATHOLDTX01 register defines the size of the four DMA thresholds for interrupt pacing for USB0. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_TX0_1 register is shown in Figure 15-30 and described in Table 15-38.

Figure 15-30. USBSS IRQ_DMA_THRESHOLD_TX0_1 Register (IRQDMATHOLDTX01)

31–24	23–16	15–8	7–0
dma_thres_tx0_7	dma_thres_tx0_6	dma_thres_tx0_5	dma_thres_tx0_4
R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-38. USBSS IRQ_DMA_THRESHOLD_TX0_1 Register (IRQDMATHOLDTX01) Field Descriptions

Bits	Field	Description
31-24	dma_thres_tx0_7	DMA threshold value for tx_pkt_cmp_0 for USB0 Endpoint 7
23-16	dma_thres_tx0_6	DMA threshold value for tx_pkt_cmp_0 for USB0 Endpoint 6
15-8	dma_thres_tx0_5	DMA threshold value for tx_pkt_cmp_0 for USB0 Endpoint 5
7-0	dma_thres_tx0_4	DMA threshold value for tx_pkt_cmp_0 for USB0 Endpoint 4

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.1.10 USBSS IRQ_DMA_THRESHOLD_TX0_2 Register (IRQDMATHOLDTX02)

The USBSS IRQDMATHOLDTX02 register defines the size of the four DMA thresholds for interrupt pacing for USB0. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_TX0_2 register is shown in Figure 15-31 and described in Table 15-39.

Figure 15-31. USBSS IRQ_DMA_THRESHOLD_TX0_2 Register (IRQDMATHOLDTX02)

31–24	23–16	15–8	7–0
dma_thres_tx0_11	dma_thres_tx0_10	dma_thres_tx0_9	dma_thres_tx0_8
R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-39. USBSS IRQ_DMA_THRESHOLD_TX0_2 Register (IRQDMATHOLDTX02) Field Descriptions

Bits	Field	Description
31-24	dma_thres_tx0_11	DMA threshold value for tx_pkt_cmp_0 for USB0 Endpoint 11
23-16	dma_thres_tx0_10	DMA threshold value for tx_pkt_cmp_0 for USB0 Endpoint 10
15-8	dma_thres_tx0_9	DMA threshold value for tx_pkt_cmp_0 for USB0 Endpoint 9
7-0	dma_thres_tx0_8	DMA threshold value for tx_pkt_cmp_0 for USB0 Endpoint 8

15.5.1.11 USBSS IRQ_DMA_THRESHOLD_TX0_3 Register (IRQDMATHOLDTX03)

The USBSS IRQDMATHOLDTX03 register defines the size of the four DMA thresholds for interrupt pacing for USB0. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_TX0_3 register is shown in Figure 15-31 and described in Table 15-39.

Figure 15-32. USBSS IRQ_DMA_THRESHOLD_TX0_3 Register (IRQDMATHOLDTX03)

31–24	23–16	15–8	7–0
dma_thres_tx0_15	dma_thres_tx0_14	dma_thres_tx0_13	dma_thres_tx0_12
R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-40. USBSS IRQ_DMA_THRESHOLD_TX0_3 Register (IRQDMATHOLDTX03) Field Descriptions

Bits	Field	Description
31-24	dma_thres_tx0_15	DMA threshold value for tx_pkt_cmp_0 for USB0 Endpoint 15
23-16	dma_thres_tx0_14	DMA threshold value for tx_pkt_cmp_0 for USB0 Endpoint 14
15-8	dma_thres_tx0_13	DMA threshold value for tx_pkt_cmp_0 for USB0 Endpoint 13
7-0	dma_thres_tx0_12	DMA threshold value for tx_pkt_cmp_0 for USB0 Endpoint 12

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.1.12 USBSS IRQ_DMA_THRESHOLD_RX0_0 Register (IRQDMATHOLDRX00)

The USBSS IRQDMATHOLDRX00 register defines the size of the four DMA thresholds for interrupt pacing for USB0. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_RX0_0 register is shown in Figure 15-33 and described in Table 15-41.

Figure 15-33. USBSS IRQ_DMA_THRESHOLD_RX0_0 Register (IRQDMATHOLDRX00)

31–24	23–16	15–8	7–0
dma_thres_rx0_3	dma_thres_rx0_2	dma_thres_rx0_1	Reserved
R/W-0h	R/W-0h	R/W-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-41. USBSS IRQ_DMA_THRESHOLD_RX0_0 Register (IRQDMATHOLDRX00) Field Descriptions

Bits	Field	Description
31-24	dma_thres_rx0_3	DMA threshold value for rx_pkt_cmp_0 for USB0 Endpoint 3
23-16	dma_thres_rx0_2	DMA threshold value for rx_pkt_cmp_0 for USB0 Endpoint 2
15-8	dma_thres_rx0_1	DMA threshold value for rx_pkt_cmp_0 for USB0 Endpoint 1
7-0	Reserved	Always read as 0. Writes have no effect.

15.5.1.13 USBSS IRQ_DMA_THRESHOLD_RX0_1 Register (IRQDMATHOLDRX01)

The USBSS IRQDMATHOLDRX01 register defines the size of the four DMA thresholds for interrupt pacing for USB0. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_RX0_1 register is shown in Figure 15-34 and described in Table 15-42.

Figure 15-34. USBSS IRQ_DMA_THRESHOLD_RX0_1 Register (IRQDMATHOLDRX01)

31–24	23–16	15–8	7–0
dma_thres_rx0_7	dma_thres_rx0_6	dma_thres_rx0_5	dma_thres_rx0_4
R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-42. USBSS IRQ_DMA_THRESHOLD_RX0_1 Register (IRQDMATHOLDRX00) Field Descriptions

Bits	Field	Description
31-24	dma_thres_rx0_7	DMA threshold value for rx_pkt_cmp_0 for USB0 Endpoint 7
23-16	dma_thres_rx0_6	DMA threshold value for rx_pkt_cmp_0 for USB0 Endpoint 6
15-8	dma_thres_rx0_5	DMA threshold value for rx_pkt_cmp_0 for USB0 Endpoint 5
7-0	dma_thres_rx0_4	DMA threshold value for rx_pkt_cmp_0 for USB0 Endpoint 4

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.1.14 USBSS IRQ_DMA_THRESHOLD_RX0_2 Register (IRQDMATHOLDRX02)

The USBSS IRQ_DMA_THRESHOLD_RX0_2 register defines the size of the four DMA thresholds for interrupt pacing. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_RX0_2 register is shown in Figure 15-35 and described in Table 15-43.

Figure 15-35. USBSS IRQ_DMA_THRESHOLD_RX0_2 Register (IRQDMATHOLDRX02)

31–24	23–16	15–8	7–0
dma_thres_rx0_11	dma_thres_rx0_10	dma_thres_rx0_9	dma_thres_rx0_8
R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-43. USBSS IRQ_DMA_THRESHOLD_RX0_2 Register (IRQDMATHOLDRX02) Field Descriptions

Bits	Field	Description
31-24	dma_thres_rx0_11	DMA threshold value for rx_pkt_cmp_0 for USB0 Endpoint 11
23-16	dma_thres_rx0_10	DMA threshold value for rx_pkt_cmp_0 for USB0 Endpoint 10
15-8	dma_thres_rx0_9	DMA threshold value for rx_pkt_cmp_0 for USB0 Endpoint 9
7-0	dma_thres_rx0_8	DMA threshold value for rx_pkt_cmp_0 for USB0 Endpoint 8

15.5.1.15 USBSS IRQ_DMA_THRESHOLD_RX0_3 Register (IRQDMATHOLDRX03)

The USBSS IRQDMATHOLDRX03 register defines the size of the four DMA thresholds for interrupt pacing for USB0. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_RX0_3 register is shown in Figure 15-36 and described in Table 15-44.

Figure 15-36. USBSS IRQ_DMA_THRESHOLD_RX0_3 Register (IRQDMATHOLDRX03)

31–24	23–16	15–8	7–0
dma_thres_rx0_15	dma_thres_rx0_14	dma_thres_rx0_13	dma_thres_rx0_12
R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-44. USBSS IRQ_DMA_THRESHOLD_RX0_3 Register (IRQDMATHOLDRX03) Field Descriptions

Bits	Field	Description
31-24	dma_thres_rx0_15	DMA threshold value for rx_pkt_cmp_0 for USB0 Endpoint 15
23-16	dma_thres_rx0_14	DMA threshold value for rx_pkt_cmp_0 for USB0 Endpoint 14
15-8	dma_thres_rx0_13	DMA threshold value for rx_pkt_cmp_0 for USB0 Endpoint 13
7-0	dma_thres_rx0_12	DMA threshold value for rx_pkt_cmp_0 for USB0 Endpoint 12

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.1.16 USBSS IRQ_DMA_THRESHOLD_TX1_0 Register (IRQDMATHOLDTX10)

The USBSS IRQDMATHOLDTX10 register (IRQDMATHOLDTX10) defines the size of the four DMA thresholds for interrupt pacing for USB1. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_TX1_0 register is shown in Figure 15-37 and described in Table 15-45.

Figure 15-37. USBSS IRQ_DMA_THRESHOLD_TX1_0 Register (IRQDMATHOLDTX10)

31–24	23–16	15–8	7–0
dma_thres_rx0_3	dma_thres_rx0_2	dma_thres_rx0_1	Reserved
R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-45. USBSS IRQ_DMA_THRESHOLD_TX1_0 Register (IRQDMATHOLDTX10) Field Descriptions

Bits	Field	Description
31-24	dma_thres_tx1_3	DMA threshold value for tx_pkt_cmp_0 for USB1 Endpoint 3
23-16	dma_thres_tx1_2	DMA threshold value for tx_pkt_cmp_0 for USB1 Endpoint 2
15-8	dma_thres_tx1_1	DMA threshold value for tx_pkt_cmp_0 for USB1 Endpoint 1
7-0	Reserved	Always read as 0. Writes have no effect.

15.5.1.17 USBSS IRQ_DMA_THRESHOLD_TX1_1 Register (IRQDMATHOLDTX11)

The USBSS IRQDMATHOLDTX11 register defines the size of the four DMA thresholds for interrupt pacing for USB1. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_TX1_1 register is shown in Figure 15-38 and described in Table 15-46.

Figure 15-38. USBSS IRQ_DMA_THRESHOLD_TX1_1 Register (IRQDMATHOLDTX11)

31–24	23–16	15–8	7–0
dma_thres_tx1_7	dma_thres_tx1_6	dma_thres_tx1_5	dma_thres_tx1_4
R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-46. USBSS IRQ_DMA_THRESHOLD_TX1_1 Register (IRQDMATHOLDTX11) Field Descriptions

Bits	Field	Description
31-24	dma_thres_tx1_7	DMA threshold value for tx_pkt_cmp_0 for USB1 Endpoint 7
23-16	dma_thres_tx1_6	DMA threshold value for tx_pkt_cmp_0 for USB1 Endpoint 6
15-8	dma_thres_tx1_5	DMA threshold value for tx_pkt_cmp_0 for USB1 Endpoint 5
7-0	dma_thres_tx1_4	DMA threshold value for tx_pkt_cmp_0 for USB1 Endpoint 4

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.1.18 USBSS IRQ_DMA_THRESHOLD_TX1_2 Register (IRQDMATHOLDTX12)

The USBSS IRQDMATHOLDTX12 register defines the size of the four DMA thresholds for interrupt pacing for USB1. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_TX1_2 register is shown in Figure 15-39 and described in Table 15-47.

Figure 15-39. USBSS IRQ_DMA_THRESHOLD_TX1_2 Register (IRQDMATHOLDTX12)

31–24	23–16	15–8	7–0
dma_thres_tx1_11	dma_thres_tx1_10	dma_thres_tx1_9	dma_thres_tx1_8
R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-47. USBSS IRQ_DMA_THRESHOLD_TX1_2 Register (IRQDMATHOLDTX12) Field Descriptions

Bits	Field	Description
31-24	dma_thres_tx1_11	DMA threshold value for tx_pkt_cmp_0 for USB1 Endpoint 11
23-16	dma_thres_tx1_10	DMA threshold value for tx_pkt_cmp_0 for USB1 Endpoint 10
15-8	dma_thres_tx1_9	DMA threshold value for tx_pkt_cmp_0 for USB1 Endpoint 9
7-0	dma_thres_tx1_8	DMA threshold value for tx_pkt_cmp_0 for USB1 Endpoint 8

15.5.1.19 USBSS IRQ_DMA_THRESHOLD_TX1_3 Register (IRQDMATHOLDTX13)

The USBSS IRQDMATHOLDTX13 register defines the size of the four DMA thresholds for interrupt pacing for USB1. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_TX1_3 register is shown in Figure 15-40 and described in Table 15-48.

Figure 15-40. USBSS IRQ_DMA_THRESHOLD_TX1_3 Register (IRQDMATHOLDTX13)

31–24	23–16	15–8	7–0
dma_thres_tx1_15	dma_thres_tx1_14	dma_thres_tx1_13	dma_thres_tx1_12
R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-48. USBSS IRQ_DMA_THRESHOLD_TX1_3 Register (IRQDMATHOLDTX13) Field Descriptions

Bits	Field	Description
31-24	dma_thres_tx1_15	DMA threshold value for tx_pkt_cmp_0 for USB1 Endpoint 15
23-16	dma_thres_tx1_14	DMA threshold value for tx_pkt_cmp_0 for USB1 Endpoint 14
15-8	dma_thres_tx1_13	DMA threshold value for tx_pkt_cmp_0 for USB1 Endpoint 13
7-0	dma_thres_tx1_12	DMA threshold value for tx_pkt_cmp_0 for USB1 Endpoint 12

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.1.20 USBSS IRQ_DMA_THRESHOLD_RX1_0 Register (IRQDMATHOLDRX10)

The USBSS IRQDMATHOLDRX10 register defines the size of the four DMA thresholds for interrupt pacing for USB1. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_RX1_0 register is shown in Figure 15-41 and described in Table 15-49.

Figure 15-41. USBSS IRQ_DMA_THRESHOLD_RX1_0 Register (IRQDMATHOLDRX10)

31–24	23–16	15–8	7–0
dma_thres_rx1_3	dma_thres_rx1_2	dma_thres_rx1_1	Reserved
R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-49. USBSS IRQ_DMA_THRESHOLD_RX1_0 Register (IRQDMATHOLDRX10) Field Descriptions

Bits	Field	Description
31-24	dma_thres_rx1_3	DMA threshold value for rx_pkt_cmp_0 for USB1 Endpoint 3
23-16	dma_thres_rx1_2	DMA threshold value for rx_pkt_cmp_0 for USB1 Endpoint 2
15-8	dma_thres_rx1_1	DMA threshold value for rx_pkt_cmp_0 for USB1 Endpoint 1
7-0	Reserved	Always read as 0. Writes have no effect.

15.5.1.21 USBSS IRQ_DMA_THRESHOLD_RX1_1 Register (IRQDMATHOLDRX11)

The USBSS IRQDMATHOLDRX11 register defines the size of the four DMA thresholds for interrupt pacing for USB1. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_RX1_1 register is shown in Figure 15-42 and described in Table 15-50.

Figure 15-42. USBSS IRQ_DMA_THRESHOLD_RX1_1 Register (IRQDMATHOLDRX11)

31–24	23–16	15–8	7–0
dma_thres_rx1_7	dma_thres_rx1_6	dma_thres_rx1_5	dma_thres_rx1_4
R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-50. USBSS IRQ_DMA_THRESHOLD_RX1_1 Register (IRQDMATHOLDRX11) Field Descriptions

Bits	Field	Description
31-24	dma_thres_rx1_7	DMA threshold value for rx_pkt_cmp_0 for USB1 Endpoint 7
23-16	dma_thres_rx1_6	DMA threshold value for rx_pkt_cmp_0 for USB1 Endpoint 6
15-8	dma_thres_rx1_5	DMA threshold value for rx_pkt_cmp_0 for USB1 Endpoint 5
7-0	dma_thres_rx1_4	DMA threshold value for rx_pkt_cmp_0 for USB1 Endpoint 4

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.1.22 USBSS IRQ_DMA_THRESHOLD_RX1_2 Register (IRQDMATHOLDRX12)

The USBSS IRQDMATHOLDRX12 register defines the size of the four DMA thresholds for interrupt pacing for USB1. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_RX1_2 register is shown in Figure 15-43 and described in Table 15-51.

Figure 15-43. USBSS IRQ_DMA_THRESHOLD_RX1_2 Register (IRQDMATHOLDRX12)

31–24	23–16	15–8	7–0
dma_thres_rx1_11	dma_thres_rx1_10	dma_thres_rx1_9	dma_thres_rx1_8
R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-51. USBSS IRQ_DMA_THRESHOLD_RX1_2 Register (IRQDMATHOLDRX12) Field Descriptions

Bits	Field	Description
31-24	dma_thres_rx1_11	DMA threshold value for rx_pkt_cmp_0 for USB1 Endpoint 11
23-16	dma_thres_rx1_10	DMA threshold value for rx_pkt_cmp_0 for USB1 Endpoint 10
15-8	dma_thres_rx1_9	DMA threshold value for rx_pkt_cmp_0 for USB1 Endpoint 9
7-0	dma_thres_rx1_8	DMA threshold value for rx_pkt_cmp_0 for USB1 Endpoint 8

15.5.1.23 USBSS IRQ_DMA_THRESHOLD_RX1_3 Register (IRQDMATHOLDRX13)

The USBSS IRQDMATHOLDRX13 register defines the size of the four DMA thresholds for interrupt pacing for USB1. Each threshold contains an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_DMA_THRESHOLD_RX1_3 register is shown in Figure 15-44 and described in Table 15-52.

Figure 15-44. USBSS IRQ_DMA_THRESHOLD_RX1_3 Register (IRQDMATHOLDRX13)

31–24	23–16	15–8	7–0
dma_thres_rx1_15	dma_thres_rx1_14	dma_thres_rx1_13	dma_thres_rx1_12
R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-52. USBSS IRQ_DMA_THRESHOLD_RX1_3 Register (IRQDMATHOLDRX13) Field Descriptions

Bits	Field	Description
31-24	dma_thres_rx1_15	DMA threshold value for rx_pkt_cmp_0 for USB1 Endpoint 15
23-16	dma_thres_rx1_14	DMA threshold value for rx_pkt_cmp_0 for USB1 Endpoint 14
15-8	dma_thres_rx1_13	DMA threshold value for rx_pkt_cmp_0 for USB1 Endpoint 13
7-0	dma_thres_rx1_12	DMA threshold value for rx_pkt_cmp_0 for USB1 Endpoint 12

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.1.24 USBSS IRQ_DMA_ENABLE_0 Register (IRQDMAENABLE0)

The USBSS IRQDMAENABLE0 contains 32 enables. Sixteen enables are assigned to the 16 endpoints associated with the tx_pkt_cmp_0 interrupt and 16 enables are assigned to the 16 endpoints associated with the rx_pkt_cmp_0 interrupt. Endpoint 0 for both the tx and rx does not exist; but it is included for completeness of this register.

If the enable is set; then that specific endpoint can generate an interrupt if the event pending count has exceeded the threshold.

If the enable is cleared; then that specific endpoint cannot generate an interrupt regardless if the event pending count has exceeded the threshold or not.

The USBSS IRQ_DMA_ENABLE_0 Register is shown in Figure 15-45 and described in Table 15-53.

Figure 15-45. USBSS IRQ_DMA_ENABLE_0 Register (IRQDMAENABLE0)

31	30	29	28	27	26	25	24
dma_en_rx0_15	dma_en_rx0_14	dma_en_rx0_13	dma_en_rx0_12	dma_en_rx0_11	dma_en_rx0_10	dma_en_rx0_9	dma_en_rx0_8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
dma_en_rx0_7	dma_en_rx0_6	dma_en_rx0_5	dma_en_rx0_4	dma_en_rx0_3	dma_en_rx0_2	dma_en_rx0_1	Reserved
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h
15	14	13	12	11	10	9	8
dma_en_tx0_15	dma_en_tx0_14	dma_en_tx0_13	dma_en_tx0_12	dma_en_tx0_11	dma_en_tx0_10	dma_en_tx0_9	dma_en_tx0_8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
dma_en_tx0_7	dma_en_tx0_6	dma_en_tx0_5	dma_en_tx0_4	dma_en_tx0_3	dma_en_tx0_2	dma_en_tx0_1	Reserved
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-53. USBSS IRQ_DMA_ENABLE_0 Register (IRQDMAENABLE0) Field Descriptions

Bits	Field	Description
31	dma_en_rx0_15	DMA threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 15
30	dma_en_rx0_14	DMA threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 14
29	dma_en_rx0_13	DMA threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 13
28	dma_en_rx0_12	DMA threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 12
27	dma_en_rx0_11	DMA threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 11
26	dma_en_rx0_10	DMA threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 10
25	dma_en_rx0_9	DMA threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 9
24	dma_en_rx0_8	DMA threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 8
23	dma_en_rx0_7	DMA threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 7
22	dma_en_rx0_6	DMA threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 6
21	dma_en_rx0_5	DMA threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 5
20	dma_en_rx0_4	DMA threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 4
19	dma_en_rx0_3	DMA threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 3
18	dma_en_rx0_2	DMA threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 2
17	dma_en_rx0_1	DMA threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 1
16	Reserved	Always read as 0. Writes have no effect.
15	dma_en_tx0_15	DMA threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 15
14	dma_en_tx0_14	DMA threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 14
13	dma_en_tx0_13	DMA threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 13
12	dma_en_tx0_12	DMA threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 12
11	dma_en_tx0_11	DMA threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 11
10	dma_en_tx0_10	DMA threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 10

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-53. USBSS IRQ_DMA_ENABLE_0 Register (IRQDMAENABLE0) Field Descriptions (continued)

Bits	Field	Description
9	dma_en_tx0_9	DMA threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 9
8	dma_en_tx0_8	DMA threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 8
7	dma_en_tx0_7	DMA threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 7
6	dma_en_tx0_6	DMA threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 6
5	dma_en_tx0_5	DMA threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 5
4	dma_en_tx0_4	DMA threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 4
3	dma_en_tx0_3	DMA threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 3
2	dma_en_tx0_2	DMA threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 2
1	dma_en_tx0_1	DMA threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 1
0	Reserved	Always read as 0. Writes have no effect.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.1.25 USBSS IRQ_DMA_ENABLE_1 Register (IRQDMAENABLE1)

The USBSS IRQDMAENABLE1 register contains 32 enables. Sixteen enables are assigned to the 16 endpoints associated with the tx_pkt_cmp_1 interrupt and 16 enables are assigned to the 16 endpoints associated with the rx_pkt_cmp_1 interrupt. Endpoint 0 for both the tx and rx does not exist; but it is included for completeness of this register.

If the enable is set; then that specific endpoint can generate an interrupt if the event pending count has exceeded the threshold.

If the enable is cleared; then that specific endpoint cannot generate an interrupt regardless if the event pending count has exceeded the threshold or not.

The USBSS IRQ_DMA_ENABLE_0 Register is shown in Figure 15-46 and described in Table 15-54.

Figure 15-46. USBSS IRQ_DMA_ENABLE_1 Register (IRQDMAENABLE1)

31	30	29	28	27	26	25	24
dma_en_rx1_15	dma_en_rx1_14	dma_en_rx1_13	dma_en_rx1_12	dma_en_rx1_11	dma_en_rx1_10	dma_en_rx1_9	dma_en_rx1_8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
dma_en_rx1_7	dma_en_rx1_6	dma_en_rx1_5	dma_en_rx1_4	dma_en_rx1_3	dma_en_rx1_2	dma_en_rx1_1	Reserved
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h
15	14	13	12	11	10	9	8
dma_en_tx1_15	dma_en_tx1_14	dma_en_tx1_13	dma_en_tx1_12	dma_en_tx1_11	dma_en_tx1_10	dma_en_tx1_9	dma_en_tx1_8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
dma_en_tx1_7	dma_en_tx1_6	dma_en_tx1_5	dma_en_tx1_4	dma_en_tx1_3	dma_en_tx1_2	dma_en_tx1_1	Reserved
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-54. USBSS IRQ_DMA_ENABLE_1 Register (IRQDMAENABLE1) Field Descriptions

Bits	Field	Description
31	dma_en_rx1_15	DMA threshold enable value for rx_pkt_cmp_0 for USB1 Endpoint 15
30	dma_en_rx1_14	DMA threshold enable value for rx_pkt_cmp_0 for USB1 Endpoint 14
29	dma_en_rx1_13	DMA threshold enable value for rx_pkt_cmp_0 for USB1 Endpoint 13
28	dma_en_rx1_12	DMA threshold enable value for rx_pkt_cmp_0 for USB1 Endpoint 12
27	dma_en_rx1_11	DMA threshold enable value for rx_pkt_cmp_0 for USB1 Endpoint 11
26	dma_en_rx1_10	DMA threshold enable value for rx_pkt_cmp_0 for USB1 Endpoint 10
25	dma_en_rx1_9	DMA threshold enable value for rx_pkt_cmp_0 for USB1 Endpoint 9
24	dma_en_rx1_8	DMA threshold enable value for rx_pkt_cmp_0 for USB1 Endpoint 8
23	dma_en_rx1_7	DMA threshold enable value for rx_pkt_cmp_0 for USB1 Endpoint 7
22	dma_en_rx1_6	DMA threshold enable value for rx_pkt_cmp_0 for USB1 Endpoint 6
21	dma_en_rx1_5	DMA threshold enable value for rx_pkt_cmp_0 for USB1 Endpoint 5
20	dma_en_rx1_4	DMA threshold enable value for rx_pkt_cmp_0 for USB1 Endpoint 4
19	dma_en_rx1_3	DMA threshold enable value for rx_pkt_cmp_0 for USB1 Endpoint 3
18	dma_en_rx1_2	DMA threshold enable value for rx_pkt_cmp_0 for USB1 Endpoint 2
17	dma_en_rx1_1	DMA threshold enable value for rx_pkt_cmp_0 for USB1 Endpoint 1
16	Reserved	Always read as 0. Writes have no effect.
15	dma_en_tx1_15	DMA threshold enable value for tx_pkt_cmp_0 for USB1 Endpoint 15
14	dma_en_tx1_14	DMA threshold enable value for tx_pkt_cmp_0 for USB1 Endpoint 14
13	dma_en_tx1_13	DMA threshold enable value for tx_pkt_cmp_0 for USB1 Endpoint 13
12	dma_en_tx1_12	DMA threshold enable value for tx_pkt_cmp_0 for USB1 Endpoint 12
11	dma_en_tx1_11	DMA threshold enable value for tx_pkt_cmp_0 for USB1 Endpoint 11
10	dma_en_tx1_10	DMA threshold enable value for tx_pkt_cmp_0 for USB1 Endpoint 10

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-54. USBSS IRQ_DMA_ENABLE_1 Register (IRQDMAENABLE1) Field Descriptions (continued)

Bits	Field	Description
9	dma_en_tx1_9	DMA threshold enable value for tx_pkt_cmp_0 for USB1 Endpoint 9
8	dma_en_tx1_8	DMA threshold enable value for tx_pkt_cmp_0 for USB1 Endpoint 8
7	dma_en_tx1_7	DMA threshold enable value for tx_pkt_cmp_0 for USB1 Endpoint 7
6	dma_en_tx1_6	DMA threshold enable value for tx_pkt_cmp_0 for USB1 Endpoint 6
5	dma_en_tx1_5	DMA threshold enable value for tx_pkt_cmp_0 for USB1 Endpoint 5
4	dma_en_tx1_4	DMA threshold enable value for tx_pkt_cmp_0 for USB1 Endpoint 4
3	dma_en_tx1_3	DMA threshold enable value for tx_pkt_cmp_0 for USB1 Endpoint 3
2	dma_en_tx1_2	DMA threshold enable value for tx_pkt_cmp_0 for USB1 Endpoint 2
1	dma_en_tx1_1	DMA threshold enable value for tx_pkt_cmp_0 for USB1 Endpoint 1
0	Reserved	Always read as 0. Writes have no effect.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.1.26 USBSS IRQ_FRAME_THRESHOLD_TX0_0 Register (IRQFRAMETHOLD00)

The USBSS IRQFRAMETHOLDTX00 register defines the size of the four FRAME thresholds for interrupt pacing for USB0. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_TX0_0 register is shown in Figure 15-47 and described in Table 15-55.

Figure 15-47. USBSS IRQ_FRAME_THRESHOLD_TX0_0 Register (IRQFRAMETHOLD00)

31–24	23–16	15–8	7–0
frame_thres_tx1_3	frame_thres_tx1_2	frame_thres_tx1_1	Reserved
R/W-0h	R/W-0h	R/W-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-55. USBSS IRQ_FRAME_THRESHOLD_TX0_0 Register (IRQFRAMETHOLD00) Field Descriptions

Bits	Field	Description
31-24	frame_thres_tx1_3	FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 3
23-16	frame_thres_tx1_2	FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 2
15-8	frame_thres_tx1_1	FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 1
7-0	Reserved	Always read as 0. Writes have no effect.

15.5.1.27 USBSS IRQ_FRAME_THRESHOLD_TX0_1 Register (IRQFRAMETHOLDTX01)

The USBSS IRQFRAMETHOLDTX01 register defines the size of the four FRAME thresholds for interrupt pacing for USB0. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_TX0_1 register is shown in Figure 15-48 and described in Table 15-56.

Figure 15-48. USBSS IRQ_FRAME_THRESHOLD_TX0_1 Register (IRQFRAMETHOLD01)

31–24	23–16	15–8	7–0
frame_thres_tx1_7	frame_thres_tx1_6	frame_thres_tx1_5	frame_thres_tx1_4
R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-56. USBSS IRQ_FRAME_THRESHOLD_TX0_1 Register (IRQFRAMETHOLDTX01) Field Descriptions

Bits	Field	Description
31-24	frame_thres_tx1_7	FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 7
23-16	frame_thres_tx1_6	FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 6
15-8	frame_thres_tx1_5	FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 5
7-0	frame_thres_tx1_4	FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 4

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.1.28 USBSS IRQ_FRAME_THRESHOLD_TX0_2 Register (IRQFRAMETHOLDTX02)

The USBSS IRQFRAMETHOLDTX02 register defines the size of the four FRAME thresholds for interrupt pacing for USB0. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_TX0_2 register is shown in Figure 15-49 and described in Table 15-57.

Figure 15-49. USBSS IRQ_FRAME_THRESHOLD_TX0_2 Register (IRQFRAMETHOLD02)

31–24	23–16	15–8	7–0
frame_thres_tx1_11	frame_thres_tx1_10	frame_thres_tx1_9	frame_thres_tx1_8
R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-57. USBSS IRQ_FRAME_THRESHOLD_TX0_2 Register (IRQFRAMETHOLDTX02) Field Descriptions

Bits	Field	Description
31-24	frame_thres_tx1_11	FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 11
23-16	frame_thres_tx1_10	FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 10
15-8	frame_thres_tx1_9	FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 9
7-0	frame_thres_tx1_8	FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 8

15.5.1.29 USBSS IRQ_FRAME_THRESHOLD_TX0_3 Register (IRQFRAMETHOLDTX03)

The USBSS IRQFRAMETHOLDTX03 register defines the size of the four FRAME thresholds for interrupt pacing for USB0. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_TX0_3 register is shown in Figure 15-50 and described in Table 15-58.

Figure 15-50. USBSS IRQ_FRAME_THRESHOLD_TX0_3 Register (IRQFRAMETHOLD03)

31–24	23–16	15–8	7–0
frame_thres_tx1_15	frame_thres_tx1_14	frame_thres_tx1_13	frame_thres_tx1_12
R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-58. USBSS IRQ_FRAME_THRESHOLD_TX0_3 Register (IRQFRAMETHOLDTX03) Field Descriptions

Bits	Field	Description
31-24	frame_thres_tx1_15	FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 15
23-16	frame_thres_tx1_14	FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 14
15-8	frame_thres_tx1_13	FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 13
7-0	frame_thres_tx1_12	FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 12

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.1.30 USBSS IRQ_FRAME_THRESHOLD_RX0_0 Register (IRQFRAMETHOLDRX00)

The USBSS IRQFRAMETHOLDRX00 register defines the size of the four FRAME thresholds for interrupt pacing for USB0. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_RX0_0 register is shown in Figure 15-51 and described in Table 15-59.

Figure 15-51. USBSS IRQ_FRAME_THRESHOLD_RX0_0 Register (IRQFRAMETHOLDRX00)

31–24	23–16	15–8	7–0
frame_thres_rx1_3	frame_thres_rx1_2	frame_thres_rx1_1	Reserved
R/W-0h	R/W-0h	R/W-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-59. USBSS IRQ_FRAME_THRESHOLD_RX0_0 Register (IRQFRAMETHOLDRX00) Field Descriptions

Bits	Field	Description
31-24	frame_thres_rx1_3	FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 3
23-16	frame_thres_rx1_2	FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 2
15-8	frame_thres_rx1_1	FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 1
7-0	Reserved	Always read as 0. Writes have no effect.

15.5.1.31 USBSS IRQ_FRAME_THRESHOLD_RX0_1 Register (IRQFRAMETHOLDRX01)

The USBSS IRQFRAMETHOLDRX01 register defines the size of the four FRAME thresholds for interrupt pacing for USB0. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_RX0_1 register is shown in Figure 15-52 and described in Table 15-60.

Figure 15-52. USBSS IRQ_FRAME_THRESHOLD_RX0_1 Register (IRQFRAMETHOLDRX01)

31–24	23–16	15–8	7–0
frame_thres_rx1_7	frame_thres_rx1_6	frame_thres_rx1_5	frame_thres_rx1_4
R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-60. USBSS IRQ_FRAME_THRESHOLD_RX0_1 Register (IRQFRAMETHOLDRX01) Field Descriptions

Bits	Field	Description
31-24	frame_thres_rx1_7	FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 7
23-16	frame_thres_rx1_6	FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 6
15-8	frame_thres_rx1_5	FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 5
7-0	frame_thres_rx1_4	FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 4

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.1.32 USBSS IRQ_FRAME_THRESHOLD_RX0_2 Register (IRQFRAMETHOLDRX02)

The USBSS IRQFRAMETHOLDRX02 register defines the size of the four FRAME thresholds for interrupt pacing for USB0. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_RX0_2 register is shown in Figure 15-52 and described in Table 15-61.

Figure 15-53. USBSS IRQ_FRAME_THRESHOLD_RX0_2 Register (IRQFRAMETHOLDRX02)

31–24	23–16	15–8	7–0
frame_thres_rx1_11	frame_thres_rx1_10	frame_thres_rx1_9	frame_thres_rx1_8
R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-61. USBSS IRQ_FRAME_THRESHOLD_RX0_2 Register (IRQFRAMETHOLDRX02) Field Descriptions

Bits	Field	Description
31-24	frame_thres_rx1_11	FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 11
23-16	frame_thres_rx1_10	FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 10
15-8	frame_thres_rx1_9	FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 9
7-0	frame_thres_rx1_8	FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 8

15.5.1.33 USBSS IRQ_FRAME_THRESHOLD_RX0_3 Register (IRQFRAMETHOLDRX03)

The USBSS IRQFRAMETHOLDRX03 register defines the size of the four FRAME thresholds for interrupt pacing for USB0. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_RX0_3 register is shown in Figure 15-54 and described in Table 15-62.

Figure 15-54. USBSS IRQ_FRAME_THRESHOLD_RX0_3 Register (IRQFRAMETHOLDRX03)

31–24	23–16	15–8	7–0
frame_thres_rx1_15	frame_thres_rx1_14	frame_thres_rx1_13	frame_thres_rx1_12
R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-62. USBSS IRQ_FRAME_THRESHOLD_RX0_3 Register (IRQFRAMETHOLDRX03) Field Descriptions

Bits	Field	Description
31-24	frame_thres_rx1_15	FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 15
23-16	frame_thres_rx1_14	FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 14
15-8	frame_thres_rx1_13	FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 13
7-0	frame_thres_rx1_12	FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 12

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.1.34 USBSS IRQ_FRAME_THRESHOLD_TX1_0 Register (IRQFRAMETHOLD10)

The USBSS IRQFRAMETHOLDTX10 register defines the size of the four FRAME thresholds for interrupt pacing for USB1. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_TX1_0 register is shown in Figure 15-55 and described in Table 15-63.

Figure 15-55. USBSS IRQ_FRAME_THRESHOLD_TX1_0 Register (IRQFRAMETHOLD10)

31–24	23–16	15–8	7–0
frame_thres_tx1_3	frame_thres_tx1_2	frame_thres_tx1_1	Reserved
R/W-0h	R/W-0h	R/W-0h	R/0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-63. USBSS IRQ_FRAME_THRESHOLD_TX1_0 Register (IRQFRAMETHOLD10) Field Descriptions

Bits	Field	Description
31-24	frame_thres_tx1_3	FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 3
23-16	frame_thres_tx1_2	FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 2
15-8	frame_thres_tx1_1	FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 1
7-0	Reserved	Always read as 0. Writes have no effect.

15.5.1.35 USBSS IRQ_FRAME_THRESHOLD_TX1_1 Register (IRQFRAMETHOLDTX11)

The USBSS IRQFRAMETHOLDTX11 register defines the size of the four FRAME thresholds for interrupt pacing for USB1. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_TX1_1 register is shown in Figure 15-56 and described in Table 15-64.

Figure 15-56. USBSS IRQ_FRAME_THRESHOLD_TX1_1 Register (IRQFRAMETHOLD11)

31–24	23–16	15–8	7–0
frame_thres_tx1_7	frame_thres_tx1_6	frame_thres_tx1_5	frame_thres_tx1_4
R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-64. USBSS IRQ_FRAME_THRESHOLD_TX1_1 Register (IRQFRAMETHOLDTX11) Field Descriptions

Bits	Field	Description
31-24	frame_thres_tx1_7	FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 7
23-16	frame_thres_tx1_6	FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 6
15-8	frame_thres_tx1_5	FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 5
7-0	frame_thres_tx1_4	FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 4

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.1.36 USBSS IRQ_FRAME_THRESHOLD_TX1_2 Register (IRQFRAMETHOLDTX12)

The USBSS IRQFRAMETHOLDTX12 register defines the size of the four FRAME thresholds for interrupt pacing for USB1. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_TX1_2 register is shown in Figure 15-57 and described in Table 15-65.

Figure 15-57. USBSS IRQ_FRAME_THRESHOLD_TX1_2 Register (IRQFRAMETHOLD12)

31–24	23–16	15–8	7–0
frame_thres_tx1_11	frame_thres_tx1_10	frame_thres_tx1_9	frame_thres_tx1_8
R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-65. USBSS IRQ_FRAME_THRESHOLD_TX1_2 Register (IRQFRAMETHOLDTX12) Field Descriptions

Bits	Field	Description
31-24	frame_thres_tx1_11	FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 11
23-16	frame_thres_tx1_10	FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 10
15-8	frame_thres_tx1_9	FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 9
7-0	frame_thres_tx1_8	FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 8

15.5.1.37 USBSS IRQ_FRAME_THRESHOLD_TX1_3 Register (IRQFRAMETHOLDTX13)

The USBSS IRQFRAMETHOLDTX13 register defines the size of the four FRAME thresholds for interrupt pacing for USB1. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_TX1_3 register is shown in Figure 15-58 and described in Table 15-66.

Figure 15-58. USBSS IRQ_FRAME_THRESHOLD_TX1_3 Register (IRQFRAMETHOLD13)

31–24	23–16	15–8	7–0
frame_thres_tx1_15	frame_thres_tx1_14	frame_thres_tx1_13	frame_thres_tx1_12
R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-66. USBSS IRQ_FRAME_THRESHOLD_TX1_3 Register (IRQFRAMETHOLDTX13) Field Descriptions

Bits	Field	Description
31-24	frame_thres_tx1_15	FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 15
23-16	frame_thres_tx1_14	FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 14
15-8	frame_thres_tx1_13	FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 13
7-0	frame_thres_tx1_12	FRAME threshold value for tx_pkt_cmp_0 for USB1 Endpoint 12

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.1.38 USBSS IRQ_FRAME_THRESHOLD_RX1_0 Register (IRQFRAMETHOLDRX10)

The USBSS IRQFRAMETHOLDRX10 register defines the size of the four FRAME thresholds for interrupt pacing for USB1. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_RX1_0 register is shown in Figure 15-59 and described in Table 15-67.

Figure 15-59. USBSS IRQ_FRAME_THRESHOLD_RX1_0 Register (IRQFRAMETHOLDRX10)

31–24	23–16	15–8	7–0
frame_thres_rx1_3	frame_thres_rx1_2	frame_thres_rx1_1	Reserved
R/W-0h	R/W-0h	R/W-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-67. USBSS IRQ_FRAME_THRESHOLD_RX1_0 Register (IRQFRAMETHOLDRX10) Field Descriptions

Bits	Field	Description
31-24	frame_thres_rx1_3	FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 3
23-16	frame_thres_rx1_2	FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 2
15-8	frame_thres_rx1_1	FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 1
7-0	Reserved	Always read as 0. Writes have no effect.

15.5.1.39 USBSS IRQ_FRAME_THRESHOLD_RX1_1 Register (IRQFRAMETHOLDRX11)

The USBSS IRQFRAMETHOLDRX11 register defines the size of the four FRAME thresholds for interrupt pacing for USB1. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_RX1_1 register is shown in Figure 15-60 and described in Table 15-68.

Figure 15-60. USBSS IRQ_FRAME_THRESHOLD_RX1_1 Register (IRQFRAMETHOLDRX11)

31–24	23–16	15–8	7–0
frame_thres_rx1_7	frame_thres_rx1_6	frame_thres_rx1_5	frame_thres_rx1_4
R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-68. USBSS IRQ_FRAME_THRESHOLD_RX1_1 Register (IRQFRAMETHOLDRX11) Field Descriptions

Bits	Field	Description
31-24	frame_thres_rx1_7	FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 7
23-16	frame_thres_rx1_6	FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 6
15-8	frame_thres_rx1_5	FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 5
7-0	frame_thres_rx1_4	FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 4

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.1.40 USBSS IRQ_FRAME_THRESHOLD_RX1_2 Register (IRQFRAMETHOLDRX12)

The USBSS IRQFRAMETHOLDRX12 register defines the size of the four FRAME thresholds for interrupt pacing for USB1. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_RX1_2 register is shown in Figure 15-61 and described in Table 15-69.

Figure 15-61. USBSS IRQ_FRAME_THRESHOLD_RX1_2 Register (IRQFRAMETHOLDRX12)

31–24	23–16	15–8	7–0
frame_thres_rx1_11	frame_thres_rx1_10	frame_thres_rx1_9	frame_thres_rx1_8
R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-69. USBSS IRQ_FRAME_THRESHOLD_RX1_2 Register (IRQFRAMETHOLDRX12) Field Descriptions

Bits	Field	Description
31-24	frame_thres_rx1_11	FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 11
23-16	frame_thres_rx1_10	FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 10
15-8	frame_thres_rx1_9	FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 9
7-0	frame_thres_rx1_8	FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 8

15.5.1.41 USBSS IRQ_FRAME_THRESHOLD_RX1_3 Register (IRQFRAMETHOLDRX13)

The USBSS IRQFRAMETHOLDRX13 register defines the size of the four FRAME thresholds for interrupt pacing for USB1. Each threshold contains is an 8-bit unsigned number and can range from 0 to 255. A possible interrupt can be triggered if the count for that specific endpoint has exceeded the value of the threshold. The counter for the compared value is also an 8-bit unsigned number; therefore, setting the threshold to 255 prevents the possibility of a trigger.

The USBSS IRQ_FRAME_THRESHOLD_RX1_3 register is shown in Figure 15-62 and described in Table 15-70.

Figure 15-62. USBSS IRQ_FRAME_THRESHOLD_RX1_3 Register (IRQFRAMETHOLDRX13)

31–24	23–16	15–8	7–0
frame_thres_rx1_15	frame_thres_rx1_14	frame_thres_rx1_13	frame_thres_rx1_12
R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-70. USBSS IRQ_FRAME_THRESHOLD_RX1_3 Register (IRQFRAMETHOLDRX13) Field Descriptions

Bits	Field	Description
31-24	frame_thres_rx1_15	FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 15
23-16	frame_thres_rx1_14	FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 14
15-8	frame_thres_rx1_13	FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 13
7-0	frame_thres_rx1_12	FRAME threshold value for rx_pkt_cmp_0 for USB1 Endpoint 12

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.1.42 USBSS IRQ_FRAME_ENABLE_0 Register (IRQFRAMEENABLE0)

The USBSS IRQFRAMEENABLE0 register contains 32 enables for USB0. Sixteen enables are assigned to the 16 endpoints associated with the tx_pkt_cmp_0 interrupt and 16 enables are assigned to the 16 endpoints associated with the rx_pkt_cmp_0 interrupt. Endpoint 0 for both the tx and rx does not exist; but it is included for completeness of this register.

If the enable is set; then that specific endpoint can generate an interrupt if the frame count has exceeded the threshold. If the enable is not set; then no hardware interrupt will be generated for that specific endpoint. It is still possible to generate the interrupt via software.

If the enable is cleared; then that specific endpoint cannot generate an interrupt regardless if the frame count has exceeded the threshold or not.

The USBSS IRQ_FRAME_ENABLE_0 register is shown in Figure 15-63 and described in Table 15-71.

Figure 15-63. USBSS IRQ_FRAME_ENABLE_0 Register (IRQFRAMEENABLE0)

31	30	29	28	27	26	25	24
frame_en_rx0_15	frame_en_rx0_14	frame_en_rx0_13	frame_en_rx0_12	frame_en_rx0_11	frame_en_rx0_10	frame_en_rx0_9	frame_en_rx0_8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
frame_en_rx0_7	frame_en_rx0_6	frame_en_rx0_5	frame_en_rx0_4	frame_en_rx0_3	frame_en_rx0_2	frame_en_rx0_1	Reserved
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h
15	14	13	12	11	10	9	8
frame_en_tx0_15	frame_en_tx0_14	frame_en_tx0_13	frame_en_tx0_12	frame_en_tx0_11	frame_en_tx0_10	frame_en_tx0_9	frame_en_tx0_8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
frame_en_tx0_7	frame_en_tx0_6	frame_en_tx0_5	frame_en_tx0_4	frame_en_tx0_3	frame_en_tx0_2	frame_en_tx0_1	Reserved
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-71. USBSS IRQ_FRAME_ENABLE_0 Register (IRQFRAMEENABLE0) Field Descriptions

Bits	Field	Description
31	frame_en_rx0_15	FRAME threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 15
30	frame_en_rx0_14	FRAME threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 14
29	frame_en_rx0_13	FRAME threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 13
28	frame_en_rx0_12	FRAME threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 12
27	frame_en_rx0_11	FRAME threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 11
26	frame_en_rx0_10	FRAME threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 10
25	frame_en_rx0_9	FRAME threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 9
24	frame_en_rx0_8	FRAME threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 8
23	frame_en_rx0_7	FRAME threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 7
22	frame_en_rx0_6	FRAME threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 6
21	frame_en_rx0_5	FRAME threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 5
20	frame_en_rx0_4	FRAME threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 4
19	frame_en_rx0_3	FRAME threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 3
18	frame_en_rx0_2	FRAME threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 2
17	frame_en_rx0_1	FRAME threshold enable value for rx_pkt_cmp_0 for USB0 Endpoint 1
16	Reserved	Always read as 0. Writes have no effect.
15	frame_en_tx0_15	FRAME threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 15
14	frame_en_tx0_14	FRAME threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 14
13	frame_en_tx0_13	FRAME threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 13
12	frame_en_tx0_12	FRAME threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 12
11	frame_en_tx0_11	FRAME threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 11

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-71. USBSS IRQ_FRAME_ENABLE_0 Register (IRQFRAMEENABLE0) Field Descriptions (continued)

Bits	Field	Description
10	frame_en_tx0_10	FRAME threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 10
9	frame_en_tx0_9	FRAME threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 9
8	frame_en_tx0_8	FRAME threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 8
7	frame_en_tx0_7	FRAME threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 7
6	frame_en_tx0_6	FRAME threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 6
5	frame_en_tx0_5	FRAME threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 5
4	frame_en_tx0_4	FRAME threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 4
3	frame_en_tx0_3	FRAME threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 3
2	frame_en_tx0_2	FRAME threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 2
1	frame_en_tx0_1	FRAME threshold enable value for tx_pkt_cmp_0 for USB0 Endpoint 1
0	Reserved	Always read as 0. Writes have no effect.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.1.43 USBSS IRQ_FRAME_ENABLE_1 Register (IRQFRAMEENABLE1)

The USBSS IRQFRAMEENABLE1 register contains 30 enables for USB1. Fifteen enables are assigned to the 15 endpoints associated with the tx_pkt_cmp_1 interrupt and 15 enables are assigned to the 15 endpoints associated with the rx_pkt_cmp_1 interrupt. Endpoint 0 for both the tx and rx does not exist; but it is included for completeness of this register.

If the enable is set; then that specific endpoint can generate an interrupt if the frame count has exceeded the threshold. If the enable is not set; then no hardware interrupt will be generated for that specific endpoint. It is still possible to generate the interrupt via software.

If the enable is cleared; then that specific endpoint cannot generate an interrupt regardless if the frame count has exceeded the threshold or not. The USBSS IRQ_FRAME_ENABLE_1 register is shown in Figure 15-64 and described in Table 15-72.

Figure 15-64. USBSS IRQ_FRAME_ENABLE_1 Register (IRQFRAMEENABLE1)

31	30	29	28	27	26	25	24
frame_en_rx1_15	frame_en_rx1_14	frame_en_rx1_13	frame_en_rx1_12	frame_en_rx1_11	frame_en_rx1_10	frame_en_rx1_9	frame_en_rx1_8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
frame_en_rx1_7	frame_en_rx1_6	frame_en_rx1_5	frame_en_rx1_4	frame_en_rx1_3	frame_en_rx1_2	frame_en_rx1_1	Reserved
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8
frame_en_tx1_15	frame_en_tx1_14	frame_en_tx1_13	frame_en_tx1_12	frame_en_tx1_11	frame_en_tx1_10	frame_en_tx1_9	frame_en_tx1_8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
frame_en_tx1_7	frame_en_tx1_6	frame_en_tx1_5	frame_en_tx1_4	frame_en_tx1_3	frame_en_tx1_2	frame_en_tx1_1	Reserved
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-72. USBSS IRQ_FRAME_ENABLE_1 Register (IRQFRAMEENABLE1) Field Descriptions

Bits	Field	Description
31	frame_en_rx1_15	FRAME threshold enable value for rx_pkt_cmp_1 for USB0 Endpoint 15
30	frame_en_rx1_14	FRAME threshold enable value for rx_pkt_cmp_1 for USB0 Endpoint 14
29	frame_en_rx1_13	FRAME threshold enable value for rx_pkt_cmp_1 for USB0 Endpoint 13
28	frame_en_rx1_12	FRAME threshold enable value for rx_pkt_cmp_1 for USB0 Endpoint 12
27	frame_en_rx1_11	FRAME threshold enable value for rx_pkt_cmp_1 for USB0 Endpoint 11
26	frame_en_rx1_10	FRAME threshold enable value for rx_pkt_cmp_1 for USB0 Endpoint 10
25	frame_en_rx1_9	FRAME threshold enable value for rx_pkt_cmp_1 for USB0 Endpoint 9
24	frame_en_rx1_8	FRAME threshold enable value for rx_pkt_cmp_1 for USB0 Endpoint 8
23	frame_en_rx1_7	FRAME threshold enable value for rx_pkt_cmp_1 for USB0 Endpoint 7
22	frame_en_rx1_6	FRAME threshold enable value for rx_pkt_cmp_1 for USB0 Endpoint 6
21	frame_en_rx1_5	FRAME threshold enable value for rx_pkt_cmp_1 for USB0 Endpoint 5
20	frame_en_rx1_4	FRAME threshold enable value for rx_pkt_cmp_1 for USB0 Endpoint 4
19	frame_en_rx1_3	FRAME threshold enable value for rx_pkt_cmp_1 for USB0 Endpoint 3
18	frame_en_rx1_2	FRAME threshold enable value for rx_pkt_cmp_1 for USB0 Endpoint 2
17	frame_en_rx1_1	FRAME threshold enable value for rx_pkt_cmp_1 for USB0 Endpoint 1
16	Reserved	Always read as 0. Writes have no effect.
15	frame_en_tx1_15	FRAME threshold enable value for tx_pkt_cmp_1 for USB0 Endpoint 15
14	frame_en_tx1_14	FRAME threshold enable value for tx_pkt_cmp_1 for USB0 Endpoint 14
13	frame_en_tx1_13	FRAME threshold enable value for tx_pkt_cmp_1 for USB0 Endpoint 13
12	frame_en_tx1_12	FRAME threshold enable value for tx_pkt_cmp_1 for USB0 Endpoint 12
11	frame_en_tx1_11	FRAME threshold enable value for tx_pkt_cmp_1 for USB0 Endpoint 11

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-72. USBSS IRQ_FRAME_ENABLE_1 Register (IRQFRAMEENABLE1) Field Descriptions (continued)

Bits	Field	Description
10	frame_en_tx1_10	FRAME threshold enable value for tx_pkt_cmp_1 for USB0 Endpoint 10
9	frame_en_tx1_9	FRAME threshold enable value for tx_pkt_cmp_1 for USB0 Endpoint 9
8	frame_en_tx1_8	FRAME threshold enable value for tx_pkt_cmp_1 for USB0 Endpoint 8
7	frame_en_tx1_7	FRAME threshold enable value for tx_pkt_cmp_1 for USB0 Endpoint 7
6	frame_en_tx1_6	FRAME threshold enable value for tx_pkt_cmp_1 for USB0 Endpoint 6
5	frame_en_tx1_5	FRAME threshold enable value for tx_pkt_cmp_1 for USB0 Endpoint 5
4	frame_en_tx1_4	FRAME threshold enable value for tx_pkt_cmp_1 for USB0 Endpoint 4
3	frame_en_tx1_3	FRAME threshold enable value for tx_pkt_cmp_1 for USB0 Endpoint 3
2	frame_en_tx1_2	FRAME threshold enable value for tx_pkt_cmp_1 for USB0 Endpoint 2
1	frame_en_tx1_1	FRAME threshold enable value for tx_pkt_cmp_1 for USB0 Endpoint 1
0	Reserved	Always read as 0. Writes have no effect.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2 USB0 and USB1 Controller Registers

The USB0 and USB1 register descriptions are included in this section.

15.5.2.1 USB0 Controller Registers

Table 15-73 lists the registers for the USB0 Controller submodule.

Table 15-73. USB0 Controller Registers

Address Offset	Acronym	USB0 Controller Register
1000h	USB0REV	USB0 REVISION
1014h	USB0CTRL	USB0 Control Register
1018h	USB0STAT	USB0 Status Register
1020h	USB0IRQMSTAT	USB0 IRQ_MERGED_STATUS
1024h	USB0IRQEOI	USB0 IRQ_EOI
1028h	USB0IRQSTATRAW0	USB0 IRQ_STATUS_RAW_0
102Ch	USB0IRQSTATRAW1	USB0 IRQ_STATUS_RAW_1
1030h	USB0IRQSTAT0	USB0 IRQ_STATUS_0
1034h	USB0IRQSTAT1	USB0 IRQ_STATUS_1
1038h	USB0IRQENABLESET0	USB0 IRQ_ENABLE_SET_0
103Ch	USB0IRQENABLESET1	USB0 IRQ_ENABLE_SET_1
1040h	USB0IRQENABLECLR0	USB0 IRQ_ENABLE_CLR_0
1044h	USB0IRQENABLECLR1	USB0 IRQ_ENABLE_CLR_1
1070h	USB0TXMODE	USB0 Tx Mode Register
1074h	USB0RXMODE	USB0 Rx Mode Register
1080h	USB0GENRNDISEP1	USB0 Generic RNDIS Size EP1
1084h	USB0GENRNDISEP2	USB0 Generic RNDIS Size EP2
1088h	USB0GENRNDISEP3	USB0 Generic RNDIS Size EP3
108Ch	USB0GENRNDISEP4	USB0 Generic RNDIS Size EP4
1090h	USB0GENRNDISEP5	USB0 Generic RNDIS Size EP5
1094h	USB0GENRNDISEP6	USB0 Generic RNDIS Size EP6
1098h	USB0GENRNDISEP7	USB0 Generic RNDIS Size EP7
109Ch	USB0GENRNDISEP8	USB0 Generic RNDIS Size EP8
10A0h	USB0GENRNDISEP9	USB0 Generic RNDIS Size EP9
10A4h	USB0GENRNDISEP10	USB0 Generic RNDIS Size EP10
10A8h	USB0GENRNDISEP11	USB0 Generic RNDIS Size EP11
10ACh	USB0GENRNDISEP12	USB0 Generic RNDIS Size EP12
10B0h	USB0GENRNDISEP13	USB0 Generic RNDIS Size EP13
10B4h	USB0GENRNDISEP14	USB0 Generic RNDIS Size EP14
10B8h	USB0GENRNDISEP15	USB0 Generic RNDIS Size EP15
10D0h	USB0AUTOREQ	USB0 Auto Req Register
10D4h	USB0SRPFIXTIME	USB0 SRP Fix Time
10D8h	USB0TDOWN	USB0 Teardown Register
10E0h	USB0UTMI	USB0 PHY UTMI Register
10E4h	USB0UTMILB	USB0 MGC UTMI Loopback Register
10E8h	USB0MODE	USB0 Mode Register
1400h–1468Ch	-	USB0 Mentor Core Registers/FIFOs
146Ch	USB0HWVERS	USB0 Mentor Core HWVERS Register (see Section 15.5.7.2.6)
0x1470 – 0x159C	-	USB0 Mentor Core Registers/FIFOs

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.1.1 USB0 Revision Register (USB0REV)

The USB0 revision register (USB0REV) register contains the major and minor revisions for the USB0 module. The USB0 revision register is shown in Figure 15-65 and described in Table 15-74.

Figure 15-65. USB0 Revision Register (USB0REV)

31	30	29	28	27
Scheme		Reserved		func
R/0-1		R/0		R/0-EA2h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R_rtl					X_major			Custom		Y_minor					
R/0-1					R/0-0			R/0-0		R/0-0					

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-74. USB0 Revision Register (USB0REV) Field Descriptions

Bits	Field	Value	Description
31-30	Scheme		Used to distinguish between old scheme and current
		0	Legacy ASP or WTBU
		1	Highlander 0.8
29-28	Reserved	0	Reserved
27-16	func		Function indicates a software compatible module family.
15-11	R_rtl		RTL revision. Will vary depending on release.
10-8	X_major		Major revision
7-6	Custom		Custom revision
5-0	Y_minor		Minor revision

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.1.2 USB0 Control Register (USB0CTRL)

The USB0 control register (USB0CTRL) allows the CPU to control various aspects of the module. If uint is set high, then the Mentor controller generic interrupt for USB[9] will be generated (if enabled). This requires software to read the Mentor controller's registers to determine which interrupt USB[0] to USB[7] occurred. If uint is set low, then the usb20otg_f module will automatically read the Mentor controller's registers and set the appropriate interrupt USB[0] to USB[7] (if enabled). The generic interrupt for USB[9] will not be generated.

The USB0 control register is shown in Figure 15-66 and described in Table 15-75.

Figure 15-66. USB0 Control Register (USB0CTRL)

31	30	29 … 16
dis_deb	dis_srp	Reserved
R/W-0h	R/W-0h	R-0h

15 … 8
Reserved
R-0h

7 … 6	5	4	3	2	1	0
Reserved	soft reset isolation	rndis	uint	Reserved	clkfack	soft reset
R-0h	R/W-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W1-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-75. USB0 Control Register (USB0CTRL) Field Descriptions

Bits	Field	Value	Description
31	dis_deb		Disable the VBUS debouncer circuit fix
30	dis_srp		Disable the OTG Session Request Protocol (SRP) AVALID circuit fix. When enabled (=0) this allows additional time for the VBUS signal to be measured against the VBUS thresholds. The time is specified in the USB0 SRP Fix Time Register.
29-6	Reserved		Always read as 0. Writes have no effect.
5	soft reset isolation		Soft reset isolation. When high this bit forces all USB0 signals that connect to the USBSS to zeros during a soft reset via bit 0 of this register. This bit should be set high prior to setting bit 0 and cleared after bit 0 is cleared.
4	rndis		Global RNDIS mode enable for all endpoints
3	uint		USB non-Highlander interrupt enable
		1	Non-Highlander
		0	Highlander
2	Reserved		Always read as 0. Writes have no effect.
1	clkfack		Clock stop fast ack enable
0	soft reset		Software reset of USB0
		Write 0	No action
		Write 1	Initiate software reset
		Read 0	Reset done, no action
		Read 1	Reset ongoing

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.1.3 USB0 Status Register (USB0STAT)

The USB0 status register (USB0STAT) allows the CPU to check the voltage state level of USB0_DRVVBUS pin. The USB0 status register is shown in Figure 15-67 and described in Table 15-76.

Figure 15-67. USB0 Status Register (USB0STAT)

31	1	0
Reserved		drvvbus
R-0h		R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-76. USB0 Status Register (USB0STAT) Field Descriptions

Bits	Field	Description
31-1	Reserved	Always read as 0. Writes have no effect.
0	drvvbus	Current USB0_DRVVBUS value

15.5.2.1.4 USB0 IRQ_MERGED_STATUS Register (USB0IRQMSTAT)

The USB0 IRQ_MERGED_STATUS register (USB0IRQMSTAT) register contains a merged status for all the events for USB0. This register optimizes software accesses in a module where a large number of events are associated to one IRQ line. The merged status is read-only, and does not need to be actively cleared like the IRQ status. It will clear automatically.

The USB0 IRQ_MERGED_STATUS register is shown in Figure 15-68 and described in Table 15-77.

Figure 15-68. USB0 IRQ_MERGED_STATUS Register (USB0IRQMSTAT)

31	2	1	0
Reserved		Bank1	Bank1
R-0h		R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-77. USB0 IRQ_MERGED_STATUS Register (USB0IRQMSTAT) Field Descriptions

Bits	Field	Unit	Description
31-2	Reserved		Always read as 0. Writes have no effect.
1	Bank1	0	No events pending from IRQ_STATUS_1
		1	At least one event is pending from IRQ_STATUS_1
0	Bank0	0	No events pending from IRQ_STATUS_0
		1	At least one event is pending from IRQ_STATUS_0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.1.5 USB0 IRQ_EOI Register (USB0IRQEOI)

The USB0 IRQ_EOI register (USB0IRQEOI) allows the CPU to acknowledge completion of an interrupt by writing to the EOI. An eoi_write signal will be generated and another interrupt will be triggered if interrupt sources remain.

This register will be reset one cycle after it has been written to.

The USB0 IRQ_EOI register is shown in Figure 15-69 and described in Table 15-78.

Figure 15-69. USB0 IRQ_EOI Register (USB0IRQEOI)

31	1	0
Reserved		EOI for USB0
R-0h		R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-78. USB0 IRQ_EOI Register (USB0IRQEOI) Field Descriptions

Bits	Field	Description
31-1	Reserved	Always read as 0. Writes have no effect.
0	EOI for USB0	EOI for USB0 interrupt

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.1.6 USB0 IRQ_STATUS_RAW_0 Register (USB0IRQSTATRAW0)

The USB0 IRQ_STATUS_RAW_0 register (USB0IRQSTATRAW0) allows the USB0 interrupt sources to be manually triggered when writing a 1 to a specific bit. A read of this register returns the USB0 interrupt event pending value.

General actions per bit:
Write 0: No action
Write 1: Set event
Read 0: No event pending
Read 1: Event pending

The USB0 IRQ_STATUS_RAW_0 register is shown in Figure 15-70 and described in Table 15-79.

Figure 15-70. USB0 IRQ_STATUS_RAW_0 Register (USB0IRQSTATRAW0)

31	30	29	28	27	26	25	24
RX EP 15	RX EP 14	RX EP 13	RX EP 12	RX EP 11	RX EP 10	RX EP 9	RX EP 8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RX EP 7	RX EP 6	RX EP 5	RX EP 4	RX EP 3	RX EP 2	RX EP 1	Reserved
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h
15	14	13	12	11	10	9	8
TX EP 15	TX EP 14	TX EP 13	TX EP 12	TX EP 11	TX EP 10	TX EP 9	TX EP 8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
TX EP 7	TX EP 6	TX EP 5	TX EP 4	TX EP 3	TX EP 2	TX EP 1	Reserved
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-79. USB0 IRQ_STATUS_RAW_0 Register (USB0IRQSTATRAW0) Field Descriptions

Bits	Field	Description
31	RX EP 15	Interrupt status for RX endpoint 15
30	RX EP 14	Interrupt status for RX endpoint 14
29	RX EP 13	Interrupt status for RX endpoint 13
28	RX EP 12	Interrupt status for RX endpoint 12
27	RX EP 11	Interrupt status for RX endpoint 11
26	RX EP 10	Interrupt status for RX endpoint 10
25	RX EP 9	Interrupt status for RX endpoint 9
24	RX EP 8	Interrupt status for RX endpoint 8
23	RX EP 7	Interrupt status for RX endpoint 7
22	RX EP 6	Interrupt status for RX endpoint 6
21	RX EP 5	Interrupt status for RX endpoint 5
20	RX EP 4	Interrupt status for RX endpoint 4
19	RX EP 3	Interrupt status for RX endpoint 3
18	RX EP 2	Interrupt status for RX endpoint 2
17	RX EP 1	Interrupt status for RX endpoint 1
16	Reserved	Always read 0. Writes have no effect.
15	TX EP 15	Interrupt status for TX endpoint 15
14	TX EP 14	Interrupt status for TX endpoint 14
13	TX EP 13	Interrupt status for TX endpoint 13
12	TX EP 12	Interrupt status for TX endpoint 12
11	TX EP 11	Interrupt status for TX endpoint 11
10	TX EP 10	Interrupt status for TX endpoint 10

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-79. USB0 IRQ_STATUS_RAW_0 Register (USB0IRQSTATRAW0) Field Descriptions (continued)

Bits	Field	Description
9	TX EP 9	Interrupt status for TX endpoint 9
8	TX EP 8	Interrupt status for TX endpoint 8
7	TX EP 7	Interrupt status for TX endpoint 7
6	TX EP 6	Interrupt status for TX endpoint 6
5	TX EP 5	Interrupt status for TX endpoint 5
4	TX EP 4	Interrupt status for TX endpoint 4
3	TX EP 3	Interrupt status for TX endpoint 3
2	TX EP 2	Interrupt status for TX endpoint 2
1	TX EP 1	Interrupt status for TX endpoint 1
0	TX EP 0	Interrupt status for TX endpoint 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.1.7 USB0 IRQ_STATUS_RAW_1 Register (USB0IRQSTATRAW1)

The USB0 IRQ_STATUS_RAW_1 register (USB0IRQSTATRAW1) allows the USB0 interrupt sources to be manually triggered when writing a 1 to a specific bit. A read of this register returns the USB0 interrupt event pending value.

General actions per bit:
Write 0: No action
Write 1: Set event
Read 0: No event pending
Read 1: Event pending

The USB0 IRQ_STATUS_RAW_1 register is shown in Figure 15-71 and described in Table 15-80.

Figure 15-71. USB0 IRQ_STATUS_RAW_1 Register (USB0IRQSTATRAW1)

31	30	29	28	27	26	25	24
TX FIFO 15	TX FIFO 14	TX FIFO 13	TX FIFO 12	TX FIFO 11	TX FIFO 10	TX FIFO 9	TX FIFO 8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
TX FIFO 7	TX FIFO 6	TX FIFO 5	TX FIFO 4	TX FIFO 3	TX FIFO 2	TX FIFO 1	TX FIFO 0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15					10	9	8
Reserved						USB[9]	USB[8]
R-0h						R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
USB[7]	USB[6]	USB[5]	USB[4]	USB[3]	USB[2]	USB[1]	USB[0]
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-80. USB0 IRQ_STATUS_RAW_1 Register (USB0IRQSTATRAW1) Field Descriptions

Bits	Field	Description
31	TX FIFO 15	Interrupt status for TX FIFO endpoint 15
30	TX FIFO 14	Interrupt status for TX FIFO endpoint 14
29	TX FIFO 13	Interrupt status for TX FIFO endpoint 13
28	TX FIFO 12	Interrupt status for TX FIFO endpoint 12
27	TX FIFO 11	Interrupt status for TX FIFO endpoint 11
26	TX FIFO 10	Interrupt status for TX FIFO endpoint 10
25	TX FIFO 9	Interrupt status for TX FIFO endpoint 9
24	TX FIFO 8	Interrupt status for TX FIFO endpoint 8
23	TX FIFO 7	Interrupt status for TX FIFO endpoint 7
22	TX FIFO 6	Interrupt status for TX FIFO endpoint 6
21	TX FIFO 5	Interrupt status for TX FIFO endpoint 5
20	TX FIFO 4	Interrupt status for TX FIFO endpoint 4
19	TX FIFO 3	Interrupt status for TX FIFO endpoint 3
18	TX FIFO 2	Interrupt status for TX FIFO endpoint 2
17	TX FIFO 1	Interrupt status for TX FIFO endpoint 1
16	TX FIFO 0	Interrupt status for TX FIFO endpoint 0
15-10	Reserved	Always read 0. Writes have no effect.
9	USB[9]	Interrupt status for Mentor controller USB_INT generic interrupt
8	USB[8]	Interrupt status for DRVVBUS level change
7	USB[7]	Interrupt status for VBUS < VBUS valid threshold
6	USB[6]	Interrupt status for SRP detected
5	USB[5]	Interrupt status for device disconnected (host mode)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-80. USB0 IRQ_STATUS_RAW_1 Register (USB0IRQSTATRAW1) Field Descriptions (continued)

Bits	Field	Description
4	USB[4]	Interrupt status for device connected (host mode)
3	USB[3]	Interrupt status for SOF started
2	USB[2]	Interrupt status for Reset signaling detected (peripheral mode) Babble detected (host mode)
1	USB[1]	Interrupt status for Resume signaling detected
0	USB[0]	Interrupt status for Suspend signaling detected

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.1.8 USB0 IRQ_STATUS_0 Register (USB0IRQSTAT0)

The USB0 IRQ_STATUS_0 register (USB0IRQSTAT0) allows the USB0 interrupt sources to be manually cleared when writing a 1 to a specific bit. A read of this register returns the USB0 interrupt event pending value.

General actions per bit:
Write 0: No action
Write 1: Clear event
Read 0: No event pending
Read 1: Event pending

The USB0 IRQ_STATUS_0 register is shown in Figure 15-72 and described in Table 15-81.

Figure 15-72. USB0 IRQ_STATUS_0 Register (USB0IRQSTAT0)

31	30	29	28	27	26	25	24
RX EP 15	RX EP 14	RX EP 13	RX EP 12	RX EP 11	RX EP 10	RX EP 9	RX EP 8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RX EP 7	RX EP 6	RX EP 5	RX EP 4	RX EP 3	RX EP 2	RX EP 1	Reserved
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h
15	14	13	12	11	10	9	8
TX EP 15	TX EP 14	TX EP 13	TX EP 12	TX EP 11	TX EP 10	TX EP 9	TX EP 8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
TX EP 7	TX EP 6	TX EP 5	TX EP 4	TX EP 3	TX EP 2	TX EP 1	Reserved
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-81. USB0 IRQ_STATUS_0 Register (USB0IRQSTAT0) Field Descriptions

Bits	Field	Description
31	RX EP 15	Interrupt status for RX endpoint 15
30	RX EP 14	Interrupt status for RX endpoint 14
29	RX EP 13	Interrupt status for RX endpoint 13
28	RX EP 12	Interrupt status for RX endpoint 12
27	RX EP 11	Interrupt status for RX endpoint 11
26	RX EP 10	Interrupt status for RX endpoint 10
25	RX EP 9	Interrupt status for RX endpoint 9
24	RX EP 8	Interrupt status for RX endpoint 8
23	RX EP 7	Interrupt status for RX endpoint 7
22	RX EP 6	Interrupt status for RX endpoint 6
21	RX EP 5	Interrupt status for RX endpoint 5
20	RX EP 4	Interrupt status for RX endpoint 4
19	RX EP 3	Interrupt status for RX endpoint 3
18	RX EP 2	Interrupt status for RX endpoint 2
17	RX EP 1	Interrupt status for RX endpoint 1
16	Reserved	Always read 0. Writes have no effect.
15	TX EP 15	Interrupt status for TX endpoint 15
14	TX EP 14	Interrupt status for TX endpoint 14
13	TX EP 13	Interrupt status for TX endpoint 13
12	TX EP 12	Interrupt status for TX endpoint 12
11	TX EP 11	Interrupt status for TX endpoint 11
10	TX EP 10	Interrupt status for TX endpoint 10

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-81. USB0 IRQ_STATUS_0 Register (USB0IRQSTAT0) Field Descriptions (continued)

Bits	Field	Description
9	TX EP 9	Interrupt status for TX endpoint 9
8	TX EP 8	Interrupt status for TX endpoint 8
7	TX EP 7	Interrupt status for TX endpoint 7
6	TX EP 6	Interrupt status for TX endpoint 6
5	TX EP 5	Interrupt status for TX endpoint 5
4	TX EP 4	Interrupt status for TX endpoint 4
3	TX EP 3	Interrupt status for TX endpoint 3
2	TX EP 2	Interrupt status for TX endpoint 2
1	TX EP 1	Interrupt status for TX endpoint 1
0	TX EP 0	Interrupt status for TX endpoint 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.1.9 USB0 IRQ_STATUS_1 Register (USB0IRQSTAT1)

The USB0 IRQ_STATUS_1 register (USB0IRQSTAT1) allows the USB0 interrupt sources to be manually cleared when writing a 1 to a specific bit. A read of this register returns the USB0 interrupt event pending value.

General actions per bit:
Write 0: No action
Write 1: Clear event
Read 0: No event pending
Read 1: Event pending

The USB0 IRQ_STATUS_1 register is shown in Figure 15-73 and described in Table 15-82.

Figure 15-73. USB0 IRQ_STATUS_1 Register (USB0IRQSTAT1)

31	30	29	28	27	26	25	24
TX FIFO 15	TX FIFO 14	TX FIFO 13	TX FIFO 12	TX FIFO 11	TX FIFO 10	TX FIFO 9	TX FIFO 8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
TX FIFO 7	TX FIFO 6	TX FIFO 5	TX FIFO 4	TX FIFO 3	TX FIFO 2	TX FIFO 1	TX FIFO 0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15					10	9	8
Reserved						USB[9]	USB[8]
R-0h						R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
USB[7]	USB[6]	USB[5]	USB[4]	USB[3]	USB[2]	USB[1]	USB[0]
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-82. USB0 IRQ_STATUS_1 Register (USB0IRQSTAT1) Field Descriptions

Bits	Field	Description
31	TX FIFO 15	Interrupt status for TX FIFO endpoint 15
30	TX FIFO 14	Interrupt status for TX FIFO endpoint 14
29	TX FIFO 13	Interrupt status for TX FIFO endpoint 13
28	TX FIFO 12	Interrupt status for TX FIFO endpoint 12
27	TX FIFO 11	Interrupt status for TX FIFO endpoint 11
26	TX FIFO 10	Interrupt status for TX FIFO endpoint 10
25	TX FIFO 9	Interrupt status for TX FIFO endpoint 9
24	TX FIFO 8	Interrupt status for TX FIFO endpoint 8
23	TX FIFO 7	Interrupt status for TX FIFO endpoint 7
22	TX FIFO 6	Interrupt status for TX FIFO endpoint 6
21	TX FIFO 5	Interrupt status for TX FIFO endpoint 5
20	TX FIFO 4	Interrupt status for TX FIFO endpoint 4
19	TX FIFO 3	Interrupt status for TX FIFO endpoint 3
18	TX FIFO 2	Interrupt status for TX FIFO endpoint 2
17	TX FIFO 1	Interrupt status for TX FIFO endpoint 1
16	TX FIFO 0	Interrupt status for TX FIFO endpoint 0
15-10	Reserved	Always read 0. Writes have no effect.
9	USB[9]	Interrupt status for Mentor controller USB_INT generic interrupt
8	USB[8]	Interrupt status for DRVVBUS level change
7	USB[7]	Interrupt status for VBUS < VBUS valid threshold
6	USB[6]	Interrupt status for SRP detected
5	USB[5]	Interrupt status for device disconnected (host mode)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-82. USB0 IRQ_STATUS_1 Register (USB0IRQSTAT1) Field Descriptions (continued)

Bits	Field	Description
4	USB[4]	Interrupt status for device connected (host mode)
3	USB[3]	Interrupt status for SOF started
2	USB[2]	Interrupt status for Reset signaling detected (peripheral mode) Babble detected (host mode)
1	USB[1]	Interrupt status for Resume signaling detected
0	USB[0]	Interrupt status for Suspend signaling detected

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.1.10 USB0 IRQ_ENABLE_SET_0 Register (USB0IRQENABLESET0)

The USB0 IRQ_ENABLE_SET_0 register (USB0IRQENABLESET0) allows the USB0 interrupt sources to be manually enabled when writing a 1 to a specific bit. A read of this register returns the USB0 interrupt enabled value. General actions per bit:
Write 0: No action
Write 1: Interrupt enabled
Read 0: Interrupt disabled
Read 1: Interrupt enabled

The USB0 IRQ_ENABLE_SET_0 register is shown in Figure 15-74 and described in Table 15-83.

Figure 15-74. USB0 IRQ_ENABLE_SET_0 Register (USB0IRQENABLESET0)

31	30	29	28	27	26	25	24
RX EP 15	RX EP 14	RX EP 13	RX EP 12	RX EP 11	RX EP 10	RX EP 9	RX EP 8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RX EP 7	RX EP 6	RX EP 5	RX EP 4	RX EP 3	RX EP 2	RX EP 1	Reserved
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h
15	14	13	12	11	10	9	8
TX EP 15	TX EP 14	TX EP 13	TX EP 12	TX EP 11	TX EP 10	TX EP 9	TX EP 8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
TX EP 7	TX EP 6	TX EP 5	TX EP 4	TX EP 3	TX EP 2	TX EP 1	Reserved
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-83. USB0 IRQ_ENABLE_SET_0 Register (USB0IRQENABLESET0) Field Descriptions

Bits	Field	Description
31	RX EP 15	Interrupt enable for RX endpoint 15
30	RX EP 14	Interrupt enable for RX endpoint 14
29	RX EP 13	Interrupt enable for RX endpoint 13
28	RX EP 12	Interrupt enable for RX endpoint 12
27	RX EP 11	Interrupt enable for RX endpoint 11
26	RX EP 10	Interrupt enable for RX endpoint 10
25	RX EP 9	Interrupt enable for RX endpoint 9
24	RX EP 8	Interrupt enable for RX endpoint 8
23	RX EP 7	Interrupt enable for RX endpoint 7
22	RX EP 6	Interrupt enable for RX endpoint 6
21	RX EP 5	Interrupt enable for RX endpoint 5
20	RX EP 4	Interrupt enable for RX endpoint 4
19	RX EP 3	Interrupt enable for RX endpoint 3
18	RX EP 2	Interrupt enable for RX endpoint 2
17	RX EP 1	Interrupt enable for RX endpoint 1
16	Reserved	Always read 0. Writes have no effect.
15	TX EP 15	Interrupt enable for TX endpoint 15
14	TX EP 14	Interrupt enable for TX endpoint 14
13	TX EP 13	Interrupt enable for TX endpoint 13
12	TX EP 12	Interrupt enable for TX endpoint 12
11	TX EP 11	Interrupt enable for TX endpoint 11
10	TX EP 10	Interrupt enable for TX endpoint 10
9	TX EP 9	Interrupt enable for TX endpoint 9

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-83. USB0 IRQ_ENABLE_SET_0 Register (USB0IRQENABLESET0) Field Descriptions (continued)

Bits	Field	Description
8	TX EP 8	Interrupt enable for TX endpoint 8
7	TX EP 7	Interrupt enable for TX endpoint 7
6	TX EP 6	Interrupt enable for TX endpoint 6
5	TX EP 5	Interrupt enable for TX endpoint 5
4	TX EP 4	Interrupt enable for TX endpoint 4
3	TX EP 3	Interrupt enable for TX endpoint 3
2	TX EP 2	Interrupt enable for TX endpoint 2
1	TX EP 1	Interrupt enable for TX endpoint 1
0	TX EP 0	Interrupt enable for TX endpoint 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.1.11 USB0 IRQ_ENABLE_SET_1 Register (USB0IRQENABLESET1)

The USB0 IRQ_ENABLE_SET_1 register (USB0IRQENABLESET1) allows the USB0 interrupt sources to be manually enabled when writing a 1 to a specific bit. A read of this register returns the USB0 interrupt enabled value. General actions per bit:
Write 0: No action
Write 1: Interrupt enabled
Read 0: Interrupt disabled
Read 1: Interrupt enabled

The USB0 IRQ_ENABLE_SET_1 register is shown in Figure 15-75 and described in Table 15-84.

Figure 15-75. USB0 IRQ_ENABLE_SET_1 Register (USB0IRQENABLESET1)

31	30	29	28	27	26	25	24
TX FIFO 15	TX FIFO 14	TX FIFO 13	TX FIFO 12	TX FIFO 11	TX FIFO 10	TX FIFO 9	TX FIFO 8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
TX FIFO 7	TX FIFO 6	TX FIFO 5	TX FIFO 4	TX FIFO 3	TX FIFO 2	TX FIFO 1	TX FIFO 0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15					10	9	8
Reserved						USB[9]	USB[8]
R-0h						R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
USB[7]	USB[6]	USB[5]	USB[4]	USB[3]	USB[2]	USB[1]	USB[0]
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-84. USB0 IRQ_ENABLE_SET_1 Register (USB0IRQENABLESET1) Field Descriptions

Bits	Field	Description
31	TX FIFO 15	Interrupt enable for TX FIFO endpoint 15
30	TX FIFO 14	Interrupt enable for TX FIFO endpoint 14
29	TX FIFO 13	Interrupt enable for TX FIFO endpoint 13
28	TX FIFO 12	Interrupt enable for TX FIFO endpoint 12
27	TX FIFO 11	Interrupt enable for TX FIFO endpoint 11
26	TX FIFO 10	Interrupt enable for TX FIFO endpoint 10
25	TX FIFO 9	Interrupt enable for TX FIFO endpoint 9
24	TX FIFO 8	Interrupt enable for TX FIFO endpoint 8
23	TX FIFO 7	Interrupt enable for TX FIFO endpoint 7
22	TX FIFO 6	Interrupt enable for TX FIFO endpoint 6
21	TX FIFO 5	Interrupt enable for TX FIFO endpoint 5
20	TX FIFO 4	Interrupt enable for TX FIFO endpoint 4
19	TX FIFO 3	Interrupt enable for TX FIFO endpoint 3
18	TX FIFO 2	Interrupt enable for TX FIFO endpoint 2
17	TX FIFO 1	Interrupt enable for TX FIFO endpoint 1
16	TX FIFO 0	Interrupt enable for TX FIFO endpoint 0
15-10	Reserved	Always read 0. Writes have no effect.
9	USB[9]	Interrupt enable for Mentor controller USB_INT generic interrupt
8	USB[8]	Interrupt enable for DRVVBUS level change
7	USB[7]	Interrupt enable for VBUS < VBUS valid threshold
6	USB[6]	Interrupt enable for SRP detected
5	USB[5]	Interrupt enable for device disconnected (host mode)
4	USB[4]	Interrupt enable for device connected (host mode)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-84. USB0 IRQ_ENABLE_SET_1 Register (USB0IRQENABLESET1) Field Descriptions (continued)

Bits	Field	Description
3	USB[3]	Interrupt enable for SOF started
2	USB[2]	Interrupt enable for Reset signaling detected (peripheral mode) Babble detected (host mode)
1	USB[1]	Interrupt enable for Resume signaling detected
0	USB[0]	Interrupt enable for Suspend signaling detected

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.1.12 USB0 IRQ_ENABLE_CLR_0 Register (USB0IRQENABLECLR0)

The USB0 IRQ_ENABLE_CLR_0 register (USB0IRQENABLECLR0) allows the USB0 interrupt sources to be manually disabled when writing a 1 to a specific bit. A read of this register returns the USB0 interrupt enabled value. General actions per bit:
Write 0: No action
Write 1: Disable interrupt
Read 0: Interrupt disabled
Read 1: Interrupt enabled

The USB0 IRQ_ENABLE_CLR_0 register is shown in Figure 15-76 and described in Table 15-85.

Figure 15-76. USB0 IRQ_ENABLE_CLR_0 Register (USB0IRQENABLECLR0)

31	30	29	28	27	26	25	24
RX EP 15	RX EP 14	RX EP 13	RX EP 12	RX EP 11	RX EP 10	RX EP 9	RX EP 8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RX EP 7	RX EP 6	RX EP 5	RX EP 4	RX EP 3	RX EP 2	RX EP 1	Reserved
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h
15	14	13	12	11	10	9	8
TX EP 15	TX EP 14	TX EP 13	TX EP 12	TX EP 11	TX EP 10	TX EP 9	TX EP 8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
TX EP 7	TX EP 6	TX EP 5	TX EP 4	TX EP 3	TX EP 2	TX EP 1	Reserved
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-85. USB0 IRQ_ENABLE_CLR_0 Register (USB0IRQENABLECLR0) Field Descriptions

Bits	Field	Description
31	RX EP 15	Interrupt enable for RX endpoint 15
30	RX EP 14	Interrupt enable for RX endpoint 14
29	RX EP 13	Interrupt enable for RX endpoint 13
28	RX EP 12	Interrupt enable for RX endpoint 12
27	RX EP 11	Interrupt enable for RX endpoint 11
26	RX EP 10	Interrupt enable for RX endpoint 10
25	RX EP 9	Interrupt enable for RX endpoint 9
24	RX EP 8	Interrupt enable for RX endpoint 8
23	RX EP 7	Interrupt enable for RX endpoint 7
22	RX EP 6	Interrupt enable for RX endpoint 6
21	RX EP 5	Interrupt enable for RX endpoint 5
20	RX EP 4	Interrupt enable for RX endpoint 4
19	RX EP 3	Interrupt enable for RX endpoint 3
18	RX EP 2	Interrupt enable for RX endpoint 2
17	RX EP 1	Interrupt enable for RX endpoint 1
16	Reserved	Always read 0. Writes have no effect.
15	TX EP 15	Interrupt enable for TX endpoint 15
14	TX EP 14	Interrupt enable for TX endpoint 14
13	TX EP 13	Interrupt enable for TX endpoint 13
12	TX EP 12	Interrupt enable for TX endpoint 12
11	TX EP 11	Interrupt enable for TX endpoint 11
10	TX EP 10	Interrupt enable for TX endpoint 10
9	TX EP 9	Interrupt enable for TX endpoint 9

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-85. USB0 IRQ_ENABLE_CLR_0 Register (USB0IRQENABLECLR0) Field Descriptions (continued)

Bits	Field	Description
8	TX EP 8	Interrupt enable for TX endpoint 8
7	TX EP 7	Interrupt enable for TX endpoint 7
6	TX EP 6	Interrupt enable for TX endpoint 6
5	TX EP 5	Interrupt enable for TX endpoint 5
4	TX EP 4	Interrupt enable for TX endpoint 4
3	TX EP 3	Interrupt enable for TX endpoint 3
2	TX EP 2	Interrupt enable for TX endpoint 2
1	TX EP 1	Interrupt enable for TX endpoint 1
0	TX EP 0	Interrupt enable for TX endpoint 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.1.13 USB0 IRQ_ENABLE_CLR_1 Register (USB0IRQENABLECLR1)

The USB0 IRQ_ENABLE_CLR_1 register (USB0IRQENABLECLR1) allows the USB0 interrupt sources to be manually disabled when writing a 1 to a specific bit. A read of this register returns the USB0 interrupt enabled value. General actions per bit:
Write 0: No action
Write 1: Disable interrupt
Read 0: Interrupt disabled
Read 1: Interrupt enabled

The USB0 IRQ_ENABLE_CLR_1 register is shown in Figure 15-77 and described in Table 15-86.

Figure 15-77. USB0 IRQ_ENABLE_CLR_1 Register (USB0IRQENABLECLR1)

31	30	29	28	27	26	25	24
TX FIFO 15	TX FIFO 14	TX FIFO 13	TX FIFO 12	TX FIFO 11	TX FIFO 10	TX FIFO 9	TX FIFO 8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

23	22	21	20	19	18	17	16
TX FIFO 7	TX FIFO 6	TX FIFO 5	TX FIFO 4	TX FIFO 3	TX FIFO 2	TX FIFO 1	TX FIFO 0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

15					10	9	8
Reserved						USB[9]	USB[8]
R-0h						R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
USB[7]	USB[6]	USB[5]	USB[4]	USB[3]	USB[2]	USB[1]	USB[0]
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-86. USB0 IRQ_ENABLE_CLR_1 Register (USB0IRQENABLECLR1) Field Descriptions

Bits	Field	Description
31	TX FIFO 15	Interrupt enable for TX FIFO endpoint 15
30	TX FIFO 14	Interrupt enable for TX FIFO endpoint 14
29	TX FIFO 13	Interrupt enable for TX FIFO endpoint 13
28	TX FIFO 12	Interrupt enable for TX FIFO endpoint 12
27	TX FIFO 11	Interrupt enable for TX FIFO endpoint 11
26	TX FIFO 10	Interrupt enable for TX FIFO endpoint 10
25	TX FIFO 9	Interrupt enable for TX FIFO endpoint 9
24	TX FIFO 8	Interrupt enable for TX FIFO endpoint 8
23	TX FIFO 7	Interrupt enable for TX FIFO endpoint 7
22	TX FIFO 6	Interrupt enable for TX FIFO endpoint 6
21	TX FIFO 5	Interrupt enable for TX FIFO endpoint 5
20	TX FIFO 4	Interrupt enable for TX FIFO endpoint 4
19	TX FIFO 3	Interrupt enable for TX FIFO endpoint 3
18	TX FIFO 2	Interrupt enable for TX FIFO endpoint 2
17	TX FIFO 1	Interrupt enable for TX FIFO endpoint 1
16	TX FIFO 0	Interrupt enable for TX FIFO endpoint 0
15-10	Reserved	Always read 0. Writes have no effect.
9	USB[9]	Interrupt enable for Mentor controller USB_INT generic interrupt
8	USB[8]	Interrupt enable for DRVVBUS level change
7	USB[7]	Interrupt enable for VBUS < VBUS valid threshold
6	USB[6]	Interrupt enable for SRP detected
5	USB[5]	Interrupt enable for device disconnected (host mode)
4	USB[4]	Interrupt enable for device connected (host mode)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-86. USB0 IRQ_ENABLE_CLR_1 Register (USB0IRQENABLECLR1) Field Descriptions (continued)

Bits	Field	Description
3	USB[3]	Interrupt enable for SOF started
2	USB[2]	Interrupt enable for Reset signaling detected (peripheral mode) Babble detected (host mode)
1	USB[1]	Interrupt enable for Resume signaling detected
0	USB[0]	Interrupt enable for Suspend signaling detected

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.1.14 USB0 Tx Mode Register (USB0TXMODE)

The USB0 Tx mode register (USB0TXMODE) register allows the CPU to individually enable RNDIS/Generic/CDC modes for each endpoint. Using the global RNDIS enable in the Control Register overrides this register and enables RNDIS mode for all endpoints.

The USB0 Tx mode register is shown in Figure 15-78 and described in Table 15-87.

Figure 15-78. USB0 Tx Mode Register (USB0TXMODE)

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
Reserved	Tx15_mode	Tx14_mode	Tx13_mode	Tx12_mode	Tx11_mode	Tx10_mode	Tx9_mode
R/0	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
Tx8_mode	Tx7_mode	Tx6_mode	Tx5_mode	Tx4_mode	Tx3_mode	Tx2_mode	Tx1_mode
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-87. USB0 Tx Mode Register (USB0TXMODE) Field Descriptions

Bits	Field	Value	Description
31-30	Reserved	0	Always read as 0. Writes have no effect.
29-28	Tx15_mode		
		00	Transparent Mode on TX endpoint 15
		01	RNDIS MODE on TX endpoint 15
		10	CDC Mode on TX endpoint 15
		11	Generic RNDIS Mode on TX endpoint 15
27-26	Tx14_mode		
		00	Transparent Mode on TX endpoint 14
		01	RNDIS MODE on TX endpoint 14
		10	CDC Mode on TX endpoint 14
		11	Generic RNDIS Mode on TX endpoint 14
25-24	Tx13_mode		
		00	Transparent Mode on TX endpoint 13
		01	RNDIS MODE on TX endpoint 13
		10	CDC Mode on TX endpoint 13
		11	Generic RNDIS Mode on TX endpoint 13
23-22	Tx12_mode		
		00	Transparent Mode on TX endpoint 12
		01	RNDIS MODE on TX endpoint 12
		10	CDC Mode on TX endpoint 12
		11	Generic RNDIS Mode on TX endpoint 12
21-20	Tx11_mode		
		00	Transparent Mode on TX endpoint 11
		01	RNDIS MODE on TX endpoint 11
		10	CDC Mode on TX endpoint 11
		11	Generic RNDIS Mode on TX endpoint 11
19-18	Tx10_mode		
		00	Transparent Mode on TX endpoint 10
		01	RNDIS MODE on TX endpoint 10
		10	CDC Mode on TX endpoint 10
		11	Generic RNDIS Mode on TX endpoint 10

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-87. USB0 Tx Mode Register (USB0TXMODE) Field Descriptions (continued)

Bits	Field	Value	Description
17-16	Tx9_mode		
		00	Transparent Mode on TX endpoint 9
		01	RNDIS MODE on TX endpoint 9
		11	CDC Mode on TX endpoint 9
		10	Generic RNDIS Mode on TX endpoint 9
15-14	Tx8_mode		
		00	Transparent Mode on TX endpoint 8
		01	RNDIS MODE on TX endpoint 8
		10	CDC Mode on TX endpoint 8
		11	Generic RNDIS Mode on TX endpoint 8
13-12	Tx7_mode		
		00	Transparent Mode on TX endpoint 7
		01	RNDIS MODE on TX endpoint 7
		10	CDC Mode on TX endpoint 7
		11	Generic RNDIS Mode on TX endpoint 7
11-10	Tx6_mode		
		00	Transparent Mode on TX endpoint 6
		01	RNDIS MODE on TX endpoint 6
		10	CDC Mode on TX endpoint 6
		11	Generic RNDIS Mode on TX endpoint 6
9-8	Tx5_mode		
		00	Transparent Mode on TX endpoint 5
		01	RNDIS MODE on TX endpoint 5
		10	CDC Mode on TX endpoint 5
		11	Generic RNDIS Mode on TX endpoint 5
7-6	Tx4_mode		
		00	Transparent Mode on TX endpoint 4
		01	RNDIS MODE on TX endpoint 4
		10	CDC Mode on TX endpoint 4
		11	Generic RNDIS Mode on TX endpoint 4
5-4	Tx3_mode		
		00	Transparent Mode on TX endpoint 3
		01	RNDIS MODE on TX endpoint 3
		10	CDC Mode on TX endpoint 3
		11	Generic RNDIS Mode on TX endpoint 3
3-2	Tx2_mode		
		00	Transparent Mode on TX endpoint 2
		01	RNDIS MODE on TX endpoint 2
		10	CDC Mode on TX endpoint 2
		11	Generic RNDIS Mode on TX endpoint 2
1-0	Tx1_mode		
		00	Transparent Mode on TX endpoint 1
		01	RNDIS MODE on TX endpoint 1
		10	CDC Mode on TX endpoint 1
		11	Generic RNDIS Mode on TX endpoint 1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.1.15 USB0 Rx Mode Register (USB0RXMODE)

The USB0 Rx mode register (USB0RXMODE) register allows the CPU to individually enable RNDIS/Generic/CDC modes for each endpoint. Using the global RNDIS enable in the control register overrides this register and enable RNDIS mode for all endpoints.

The USB0 Rx mode register is shown in Figure 15-79 and described in Table 15-88.

Figure 15-79. USB0 Rx Mode Register (USB0RXMODE)

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
Reserved	Rx15_mode	Rx14_mode	Rx13_mode	Rx12_mode	Rx11_mode	Rx10_mode	Rx9_mode
R/0	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
Rx8_mode	Rx7_mode	Rx6_mode	Rx5_mode	Rx4_mode	Rx3_mode	Rx2_mode	Rx1_mode
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-88. USB0 Rx Mode Register (USB0RXMODE) Field Descriptions

Bits	Field	Value	Description
31-30	Reserved	0	Always read as 0. Writes have no effect.
29-28	Rx15_mode		
		00	Transparent Mode on RX endpoint 15
		01	RNDIS MODE on RX endpoint 15
		11	CDC Mode on RX endpoint 15
		10	Generic RNDIS Mode on RX endpoint 15
27-26	Rx14_mode		
		00	Transparent Mode on RX endpoint 14
		01	RNDIS MODE on RX endpoint 14
		10	CDC Mode on RX endpoint 14
		11	Generic RNDIS Mode on RX endpoint 14
25-24	Rx13_mode		
		00	Transparent Mode on RX endpoint 13
		01	RNDIS MODE on RX endpoint 13
		10	CDC Mode on RX endpoint 13
		11	Generic RNDIS Mode on RX endpoint 13
23-22	Rx12_mode		
		00	Transparent Mode on RX endpoint 12
		01	RNDIS MODE on RX endpoint 12
		10	CDC Mode on RX endpoint 12
		11	Generic RNDIS Mode on RX endpoint 12
21-20	Rx11_mode		
		00	Transparent Mode on RX endpoint 11
		01	RNDIS MODE on RX endpoint 11
		10	CDC Mode on RX endpoint 11
		11	Generic RNDIS Mode on RX endpoint 11
19-18	Rx10_mode		
		00	Transparent Mode on RX endpoint 10
		01	RNDIS MODE on RX endpoint 10
		10	CDC Mode on RX endpoint 10
		11	Generic RNDIS Mode on RX endpoint 10

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-88. USB0 Rx Mode Register (USB0RXMODE) Field Descriptions (continued)

Bits	Field	Value	Description
17-16	Rx9_mode		
		00	Transparent Mode on RX endpoint 9
		01	RNDIS MODE on RX endpoint 9
		10	CDC Mode on RX endpoint 9
		11	Generic RNDIS Mode on RX endpoint 9
15-14	Rx8_mode		
		00	Transparent Mode on RX endpoint 8
		01	RNDIS MODE on RX endpoint 8
		10	CDC Mode on RX endpoint 8
		11	Generic RNDIS Mode on RX endpoint 8
13-12	Rx7_mode		
		00	Transparent Mode on RX endpoint 7
		01	RNDIS MODE on RX endpoint 7
		10	CDC Mode on RX endpoint 7
		11	Generic RNDIS Mode on RX endpoint 7
11-10	Rx6_mode		
		00	Transparent Mode on RX endpoint 6
		01	RNDIS MODE on RX endpoint 6
		10	CDC Mode on RX endpoint 6
		11	Generic RNDIS Mode on RX endpoint 6
9-8	Rx5_mode		
		00	Transparent Mode on RX endpoint 5
		01	RNDIS MODE on RX endpoint 5
		10	CDC Mode on RX endpoint 5
		11	Generic RNDIS Mode on RX endpoint 5
7-6	Rx4_mode		
		00	Transparent Mode on RX endpoint 4
		01	RNDIS MODE on RX endpoint 4
		10	CDC Mode on RX endpoint 4
		11	Generic RNDIS Mode on RX endpoint 4
5-4	Rx3_mode		
		00	Transparent Mode on RX endpoint 3
		01	RNDIS MODE on RX endpoint 3
		10	CDC Mode on RX endpoint 3
		11	Generic RNDIS Mode on RX endpoint 3
3-2	Rx2_mode		
		00	Transparent Mode on RX endpoint 2
		01	RNDIS MODE on RX endpoint 2
		10	CDC Mode on RX endpoint 2
		11	Generic RNDIS Mode on RX endpoint 2
1-0	Rx1_mode		
		00	Transparent Mode on RX endpoint 1
		01	RNDIS MODE on RX endpoint 1
		10	CDC Mode on RX endpoint 1
		11	Generic RNDIS Mode on RX endpoint 1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.1.16 USB0 Generic RNDIS EPn Size Register (USB0GENRNDISEPn)

The USB0 generic RNDIS EPn size register (USB0GENRNDISEPn) is programmed with a RNDIS packet size in bytes. When EP(n) is in Generic RNDIS mode, the received USB packets will be collected into a single CPPI packet that is completed when the number of bytes equal to the value of this register has been received, or a “short” packet is received. Note that this register must be programmed with a value that is an integer multiple of the endpoint size. The maximum value this register can be programmed with is 0x10000, or 65536. N can range from 1 to 15.

The USB0 generic RNDIS EPn size register is shown in Figure 15-80 and described in Table 15-89.

Figure 15-80. USB0 Generic RNDIS EPn Size Register (USB0GENRNDISEPn)

31 – 17	16 – 0
Reserved	Ep(n)_size
R-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-89. USB0 Generic RNDIS EPn Size Register (USB0GENRNDISEPn) Field Descriptions

Bits	Field	Description
31-17	Reserved	Always read as 0. Writes have no effect.
16-0	Ep(n)_size	Generic RNDIS packet size. n=1-15

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.1.17 USB0 Auto Req Register (USB0AUTOREQ)

The USB0 auto req register (USB0AUTOREQ) allows the CPU to enable an automatic IN token request generation for host mode RX operation per each RX endpoint. This feature has the DMA set the ReqPkt bit in the RXCSR when it clears the RxPktRdy bit after reading out a packet. The ReqPkt bit is used by the core to generate an IN token to receive data. By using this feature, the host can automatically generate an IN token after the DMA finishes receiving data and empties an endpoint buffer, thus receiving the next data packet as soon as possible from the connected device. Without this feature, the CPU will have to manually set the ReqPkt bit for every USB packet.

There are two modes that Auto Req can function: always or all except an EOP. The always mode will set the ReqPkt bit after every USB packet the DMA receives thus generating a new IN token after each USB packet. The EOP mode will set the ReqPkt bit after every USB packet that isn't an EOP (end of packet) in the CPPI descriptor. For RNDIS, CDC, and Generic RNDIS modes the auto req will stop when the EOP is received (either via a short packet for RNDIS,CDC, and Generic RNDIS or the count is reached for Generic RNDIS), making it useful for starting a large RNDIS packet and having it auto generate IN tokens until the end of the RNDIS packet. For transparent mode every USB packet is an EOP CPPI packet so the auto req never functions and acts like auto req is disabled.

The USB0 auto req register is shown in Figure 15-81 and described in Table 15-90.

Figure 15-81. USB0 Auto Req Register (USB0AUTOREQ)

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
Reserved	Rx15_autoreq	Rx14_autoreq	Rx13_autoreq	Rx12_autoreq	Rx11_autoreq	Rx10_autoreq	Rx9_autoreq
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
Rx8_autoreq	Rx7_autoreq	Rx6_autoreq	Rx5_autoreq	Rx4_autoreq	Rx3_autoreq	Rx2_autoreq	Rx1_autoreq
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-90. USB0 Auto Req Register (USB0AUTOREQ) Field Descriptions

Bits	Field	Values	Description
31-30	Reserved	0	Always read as 0. Writes have no effect.
29-28	Rx15_autoreq		RX endpoint 15 Auto Req enable
		00	No auto req
		01	Auto req on all but EOP
		10	Reserved
		11	Auto req always
27-26	Rx14_autoreq		RX endpoint 14 Auto Req enable
		00	No auto req
		01	Auto req on all but EOP
		10	Reserved
		11	Auto req always
25-24	Rx13_autoreq		RX endpoint 13 Auto Req enable
		00	No auto req
		01	Auto req on all but EOP
		10	Reserved
		11	Auto req always
23-22	Rx12_autoreq		RX endpoint 12 Auto Req enable
		00	No auto req
		01	Auto req on all but EOP
		10	Reserved
		11	Auto req always

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-90. USB0 Auto Req Register (USB0AUTOREQ) Field Descriptions (continued)

Bits	Field	Values	Description
21-20	Rx11_autoreq		RX endpoint 11 Auto Req enable
		00	No auto req
		01	Auto req on all but EOP
		10	Reserved
		11	Auto req always
19-18	Rx10_autoreq		RX endpoint 10 Auto Req enable
		00	No auto req
		01	Auto req on all but EOP
		10	Reserved
		11	Auto req always
17-16	Rx9_autoreq		RX endpoint 9 Auto Req enable
		00	No auto req
		01	Auto req on all but EOP
		10	Reserved
		11	Auto req always
15-14	Rx8_autoreq		RX endpoint 8 Auto Req enable
		00	No auto req
		01	Auto req on all but EOP
		10	Reserved
		11	Auto req always
13-12	Rx7_autoreq		RX endpoint 7 Auto Req enable
		00	No auto req
		01	Auto req on all but EOP
		10	Reserved
		11	Auto req always
11-10	Rx6_autoreq		RX endpoint 6 Auto Req enable
		00	No auto req
		01	Auto req on all but EOP
		10	Reserved
		11	Auto req always
9-8	Rx5_autoreq		RX endpoint 5 Auto Req enable
		00	No auto req
		01	Auto req on all but EOP
		10	Reserved
		11	Auto req always
7-6	Rx4_autoreq		RX endpoint 4 Auto Req enable
		00	No auto req
		01	Auto req on all but EOP
		10	Reserved
		11	Auto req always
5-4	Rx3_autoreq		RX endpoint 3 Auto Req enable
		00	No auto req
		01	Auto req on all but EOP
		10	Reserved
		11	Auto req always

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-90. USB0 Auto Req Register (USB0AUTOREQ) Field Descriptions (continued)

Bits	Field	Values	Description
3-2	Rx2_autoreq		RX endpoint 2 Auto Req enable
		00	No auto req
		01	Auto req on all but EOP
		10	Reserved
		11	Auto req always
1-0	Rx1_autoreq		RX endpoint 1 Auto Req enable
		00	No auto req
		01	Auto req on all but EOP
		10	Reserved
		11	Auto req always

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.1.18 USB0 SRP Fix Time Register (USB0SRPFIXTIME)

The USB0 SRP fix time register (USB0SRPFIXTIME) allows the CPU to configure the maximum amount of time the SRP fix logic blocks the AVAID from the PHY to the OTG core. This time allows the VBUS signal the ability to get below the thresholds and therefore remove the chance of voltage bounces which could give false threshold measurements.

The USB0 SRP fix time register is shown in Figure 15-82 and described in Table 15-91.

Figure 15-82. USB0 SRP Fix Time Register (USB0SRPFIXTIME)

31	0
srpfixtime	
R/W-0280de80h	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-91. USB0 SRP Fix Time Register (USB0SRPFIXTIME) Field Descriptions

Bits	Field	Description
31-0	srpfixtime	SRP Fix maximum time in 60 MHz cycles. Default is 700 ms.

15.5.2.1.19 USB0 Teardown Register (USB0TDOWN)

The USB0 teardown register (USB0TDOWN) controls the tearing down of rx and tx fifos in the USB controller. When a '1' is written to a valid bit in this register, the CPPI FIFO pointers for that endpoint are cleared, and the register automatically clears itself after 1 clock cycle. This register must be used in conjunction with the CPPI DMA Teardown mechanism. The host should also write the FlushFIFO bits in the TXCSR and RXCSR Mentor USB Controller registers to ensure a complete teardown of the endpoint. See the Mentor specification for details.

The USB0 teardown register is shown in Figure 15-83 and described in Table 15-92.

Figure 15-83. USB0 Teardown Register (USB0TDOWN)

31	17	16	15	0
tx_tdown		Rsvd	rx_tdown	Rsvd
R/W-0h		R-0h	R/W-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-92. USB0 Teardown Register (USB0TDOWN) Field Descriptions

Bits	Field	Description
31-17	tx_tdown	Tx Endpoint Teardown. Write '1' to corresponding bit to set. Read as '0'. Bit 31 = Endpoint 15 ... Bit 17 = Endpoint 1
16	Reserved	Always read as 0. Writes have no effect.
15-1	rx_tdown	RX Endpoint Teardown. Write '1' to corresponding bit to set. Read as '0'. Bit 15 = Endpoint 15 ... Bit 1 = Endpoint 1
0	Reserved	Always read as 0. Writes have no effect.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.1.20 USB0 PHY UTMI Register (USB0UTMI)

The USB0 PHY UTMI register (USB0UTMI) controls various PHY UTMI signals. The UTMI interface is located between the PHY module and the Mentor Graphics Controller. These signals are inputs to the PHY but are not outputs from the Mentor Graphics Controller.

It is important to set the otgdisable bit in to a high value to operate the PHY in a non-OTG mode. The USB0 PHY UTMI register is shown in Figure 15-84 and described in Table 15-93.

Figure 15-84. USB0 PHY UTMI Register (USB0UTMI)

31							24
Reserved							
R-0h							

23	22	21	20	19	18	17	16
txbitstuffen	txbitstuffenh	otgdisable	vbusvldextsel	vbusvldext	txenablen	fsxcvrowner	txvalidh
R/W-0h	R/W-0h	R/W-1h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

15							8
datainh							

7					3	2	1	0
Reserved						wordinterface	fsdataext	fsse0ext
R-0h						R/W-0h	R/W-1h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-93. USB0 PHY UTMI Register (USB0UTMI) Field Descriptions

Bits	Field	Description
31-24	Reserved	Always read as 0. Writes have no effect.
23	txbitstuffen	PHY UTMI input for signal txbitstuffen
22	txbitstuffenh	PHY UTMI input for signal txbitstuffenh
21	otgdisable	PHY UTMI input for signal otgdisable
20	vbusvldextsel	PHY UTMI input for signal vbusvldextsel
19	vbusvldext	PHY UTMI input for signal vbusvldext
18	txenablen	PHY UTMI input for signal txenablen
17	fsxcvrowner	PHY UTMI input for signal fsxcvrowner
16	txvalidh	PHY UTMI input for signal txvalidh
15-8	datainh	PHY UTMI input for signal datainh
7-3	Reserved	Always read as 0. Writes have no effect.
2	wordinterface	PHY UTMI input for signal wordinterface
1	fsdataext	PHY UTMI input for signal fsdataext
0	fsse0ext	PHY UTMI input for signal fsse0ext

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.1.21 USB0 MGC UTMI Loopback Register (USB0UTMILB)

The USB0 MGC UTMI loopback register (USB0UTMILB) contains most of the input and output UTMI signals for the Mentor Graphics Controller. The UTMI interface is located between the PHY module and the Mentor Graphics Controller.

In the loopback mode test register bits 11 to 0 control various UTMI signals that are input to the Mentor Graphics Controller.

In the loopback mode test register bits 28 to 16 observe various UTMI signals that are output from the Mentor Graphics Controller.

The USB0 MGC UTMI loopback register is shown in Figure 15-85 and described in Table 15-94.

Figure 15-85. USB0 MGC UTMI Loopback Register (USB0UTMILB)

31		29	28	27	26	25	24
Reserved			suspendm	suspendm		txvalid	xcvrsel
R-0h			R-0h	R-0h		R-0h	R-0h

23	22	21	20	19	18	17	16
xcvrsel	termsel	drvvbus	chrgvbus	dischrgvbus	dppulldown	dmpulldown	idpullup
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

15			12	11	10	9	8
Reserved				iddig	hostdiscon	sessend	avalid
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
vbusvalid	rxerror	Reserved		linestate		Reserved	
R/W-0h	R/W-0h	R-0h		R/W-1h		R-0h	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-94. USB0 MGC UTMI Loopback Register (USB0UTMILB) Field Descriptions

Bits	Field	Description
31-29	Reserved	Always read as 0. Writes have no effect.
28	suspendm	Loopback test observed value for suspendm
27-26	opmode	Loopback test observed value for opmode
25	txvalid	Loopback test observed value for txvalid
24-23	xcvrsel	Loopback test observed value for xcvrsel
22	termsel	Loopback test observed value for termsel
21	drvvbus	Loopback test observed value for drvvbus
20	chrgvbus	Loopback test observed value for chrgvbus
19	dischrgvbus	Loopback test observed value for dischrgvbus
18	dppulldown	Loopback test observed value for dppulldown
17	dmpulldown	Loopback test observed value for dmpulldown
16	idpullup	Loopback test observed value for idpullup
15-12	Reserved	Always read as 0. Writes have no effect.
11	iddig	Loopback test value for iddig
10	hostdiscon	Loopback test value for hostdiscon
9	sessend	Loopback test value for sessend
8	avalid	Loopback test value for avalid
7	vbusvalid	Loopback test value for vbusvalid
6	rxerror	Loopback test value for rxerror
5-4	Reserved	Always read as 0. Writes have no effect.
3-2	linestate	Loopback test value for linestate
1-0	Reserved	Always read as 0. Writes have no effect.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.1.22 USB0 Mode Register (USB0MODE)

The USB mode (USB0MODE) register allows the user a flexibility of controlling the role that the controller assumes. Via the setting of the USB0MODE[Bit7=Iddig_Mux], the user has the capability of allowing the control to be coming from a register field (USB0MODE[Bit8=iddig]) or from the USB_ID pin which is indirectly controlled by the cable end. if USB0MODE[iddig_mux]=0, the state of the USB_ID pin controls the role the controller assumes. However, if USBMODE[Iddig_mux]=1, then the setting USB0MODE[iddig] will control the role the controller assumes.

The loopback bit enables the loopback test. This test allows MGC0 to be connected to MGC1. It is important to set both loopback bits in both USB0/1 Mode registers. The USB0 MGC UTMI loopback register contains the various UTMI signals to be controlled and observed during loopback test.

The phy_test bit enables the phy_test mode. This test mode is intended to allow additional control of the UTMI inputs to the PHY. Currently, these inputs are drvvbus, dppulldown, dmpulldown, and idpullup. When phy_test is high, then the pin inputs for these signals control the inputs to the PHY instead of the Mentor controller outputs. The phy_test mode is not active with loopback mode is active.

When phy_test is active than the PHY inputs datainh is equal to datain. And txvalidh is equal to txvalid.

The USB0 mode register is shown in Figure 15-86 and described in Table 15-95.

Figure 15-86. USB0 Mode Register (USB0MODE)

31						16
Reserved						
R-0h						

15	9	8	7	6	2	1	0
Reserved		iddig	iddig_mux	Reserved		phy_test	loopback
R-0h		R/W-1h	R/W-0h	R-0h		R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-95. USB0 Mode Register (USB0MODE) Field Descriptions

Bits	Field	Value	Description
31-9	Reserved	0	Always read as 0. Writes have no effect.
8	iddig		MGC input value for iddig
		0	A type
		1	B type
7	iddig_mux		Multiplexer control for IDDIG signal going to the controller
		0	IDDIG is setting is controller by USB_ID Pin
		1	IDDIG is from bit8 (iddig) of this register
6-2	Reserved	0	Always read as 0. Writes have no effect.
1	phy_test		PHY test
		0	Normal mode
		1	PHY test mode
0	loopback		Loopback test mode
		0	Normal mode
		1	Loopback test mode

15.5.2.1.23 USB0 Mentor Core Registers

A description of the Mentor core registers is available in Section 15.5.6. Two instantiations of the USB core exists, USB0 and USB1. These registers reside back to back within USB0 space. The core registers that are used by USB0 subsystem are defined within offsets 1400h-159Ch, while the core registers that are used by USB1 subsystem are defined within offset 1C00h-1D9Ch.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.2 USB1 Controller Registers

Table 15-96 lists the registers for the USB1 controller submodule.

Table 15-96. USB1 Controller Registers

Address Offset	Acronym	USB1 Controller Register
1800h	USB1REV	USB1 Revision Register
1814h	USB1CTRL	USB1 Control Register
1818h	USB1STAT	USB1 Status Register
1820h	USB1IRQMSTAT	USB1 IRQ_MERGED_STATUS
1824h	USB1IRQEOI	USB1 IRQ_EOI
1828h	USB1IRQSTATRAW0	USB1 IRQ_STATUS_RAW_0
182Ch	USB1IRQSTATRAW1	USB1 IRQ_STATUS_RAW_1
1830h	USB1IRQSTAT0	USB1 IRQ_STATUS_0
1834h	USB1IRQSTAT1	USB1 IRQ_STATUS_1
1838h	USB1IRQENABLESET0	USB1 IRQ_ENABLE_SET_0
183Ch	USB1IRQENABLESET1	USB1 IRQ_ENABLE_SET_1
1840h	USB1IRQENABLECLR0	USB1 IRQ_ENABLE_CLR_0
1844h	USB1IRQENABLECLR1	USB1 IRQ_ENABLE_CLR_1
1870h	USB1TXMODE	USB1 Tx Mode Register
1874h	USB1RXMODE	USB1 Rx Mode Register
1880h	USB1GENRNDISEP1	USB1 Generic RNDIS Size EP1
1884h	USB1GENRNDISEP2	USB1 Generic RNDIS Size EP2
1888h	USB1GENRNDISEP3	USB1 Generic RNDIS Size EP3
188Ch	USB1GENRNDISEP4	USB1 Generic RNDIS Size EP4
1890h	USB1GENRNDISEP5	USB1 Generic RNDIS Size EP5
1894h	USB1GENRNDISEP6	USB1 Generic RNDIS Size EP6
1898h	USB1GENRNDISEP7	USB1 Generic RNDIS Size EP7
189Ch	USB1GENRNDISEP8	USB1 Generic RNDIS Size EP8
18A0h	USB1GENRNDISEP9	USB1 Generic RNDIS Size EP9
18A4h	USB1GENRNDISEP10	USB1 Generic RNDIS Size EP10
18A8h	USB1GENRNDISEP11	USB1 Generic RNDIS Size EP11
18ACh	USB1GENRNDISEP12	USB1 Generic RNDIS Size EP12
18B0h	USB1GENRNDISEP13	USB1 Generic RNDIS Size EP13
18B4h	USB1GENRNDISEP14	USB1 Generic RNDIS Size EP14
18B8h	USB1GENRNDISEP15	USB1 Generic RNDIS Size EP15
18D0h	USB1AUTOREQ	USB1 Auto Req Register
18D4h	USB1SRPFIXTIME	USB1 SRP Fix Time
18D8h	USB1TDOWN	USB1 Teardown Register
18E0h	USB1UTMI	USB1 PHY UTMI Register
18E4h	USB1UTMILB	USB1 MGC UTMI Loopback Register
18E8h	USB1MODE	USB1 Mode Register
1C00h–1C68h	...	USB1 Mentor Core Registers/FIFOs
1C6Ch	USB1HWVERS	USB1 Mentor Core HWVERS Register (see Section 15.5.7.2.6)
1C70h-1D9Ch	...	USB1 Mentor Core Registers/FIFOs

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.2.1 USB1 Revision Register (USB1REV)

The USB1 revision register (USB1REV) contains the major and minor revisions for the USB1 module.

The USB1 revision register is shown in Figure 15-87 and described in Table 15-97.

Figure 15-87. USB1 Revision Register (USB1REV)

31–30	29–28	27–16
Scheme	Reserved	func
R/1h	R-0h	R-ea2h

15–11	10–8	7–6	5–0
R_rtl	X_major	Custom	Y_minor
R/1h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-97. USB1 Revision Register (USB1REV) Field Descriptions

Bits	Field	Values	Description
31-30	Scheme		Used to distinguish between old scheme and current.
		0	Legacy ASP or WTBU
		1	Highlander 0.8
29-28	Reserved	0	Reserved
27-16	func		Function indicates a software compatible module family.
15-11	R_rtl		RTL revision. Will vary depending on release.
10-8	X_major		Major revision.
7-6	Custom		Custom revision
5-0	Y_minor		Minor revision.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.2.2 USB1 Control Register (USB1CTRL)

The USB1 control register (USB1CTRL) allows the CPU to control various aspects of the module. If uint is set high, then the Mentor controller generic interrupt for USB[9] will be generated (if enabled). This requires software to read the Mentor controller's registers to determine which interrupt USB[0] to USB[7] occurred. If uint is set low, then the usb20otg_f module will automatically read the Mentor controller's registers and set the appropriate interrupt USB[0] to USB[7] (if enabled). The generic interrupt for USB[9] will not be generated.

The USB1 control register is shown in Figure 15-88 and described in Table 15-98.

Figure 15-88. USB1 Control Register (USB1CTRL)

31	30	29 … 16
dis_deb	dis_srp	Reserved
R/W-0h	R/W-0h	R-0h

15 … 8
Reserved
R-0h

7 … 6	5	4	3	2	1	0
Reserved	soft reset isolation	rndis	uint	Reserved	clkfack	soft reset
R-0h	R/W-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	R/W1-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-98. USB1 Control Register (USB1CTRL) Field Descriptions

Bits	Field	Value	Description
31	dis_deb		Disable the VBUS debouncer circuit fix
30	dis_srp		Disable the OTG Session Request Protocol (SRP) AVALID circuit fix. When enabled (=0) this allows additional time for the VBUS signal to be measured against the VBUS thresholds. The time is specified in the USB1 SRP Fix Time Register.
29-06	Reserved	0	Always read as 0. Writes have no effect.
5	soft reset isolation		Soft reset isolation. When high this bit forces all USB1 signals that connect to the USBSS to zeros during a soft reset via bit 0 of this register. This bit should be set high prior to setting bit 0 and cleared after bit 0 is cleared.
4	rndis		Global RNDIS mode enable for all endpoints.
3	uint		USB non-highlander interrupt enable
		0	Highlander
		1	Non-highlander
2	Reserved	0	Always read as 0. Writes have no effect.
1	clkfack		Clock stop fast ack enable.
0	soft reset		Software reset of USB1.
		Write 0	Write 0 = no action
		Write 1	Write 1 = Initiate software reset
			Read 0 = Reset done, no action
			Read 1 = Reset ongoing

If uint is set high, then the Mentor controller generic interrupt for USB[9] will be generated (if enabled). This requires software to read the Mentor controller's registers to determine which interrupt USB[0] to USB[7] occurred.

If uint is set low, then the usb20otg_f module will automatically read the Mentor controller's registers and set the appropriate interrupt USB[0] to USB[7] (if enabled). The generic interrupt for USB[9] will not be generated.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.2.3 USB1 Status Register (USB1STAT)

The USB1 status register (USB1STAT) allows the CPU to check the voltage state level of USB1_DRVVBUS pin.

Ths USB1 status register is shown in Figure 15-89 and described in Table 15-99.

Figure 15-89. USB1 Status Register (USB1STAT)

31	1	0
Reserved		drvvbus
R-0h		R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-99. USB1 Status Register (USB1STAT) Field Descriptions

Bits	Field	Description
31-1	Reserved	Always read as 0. Writes have no effect.
0	drvvbus	Current USB1_DRVVBUS value

15.5.2.2.4 USB1 IRQ_MERGED_STATUS Register (USB1IRQMSTAT)

The USB1 IRQ_MERGED_STATUS register (USB1IRQMSTAT) contains a merged status for all the events for USB1. This register optimizes software accesses in a module where a large number of events are associated to one IRQ line. The merged status is read-only, and does not need to be actively cleared like the IRQ status. It will clear automatically.

The USB1 IRQ_MERGED_STATUS register is shown in Figure 15-90 and described in Table 15-100.

Figure 15-90. USB1 IRQ_MERGED_STATUS Register (USB1IRQMSTAT)

31	2	1	0
Reserved		Bank1	Bank0
R-0h		R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-100. USB1 IRQ_MERGED_STATUS Register (USB1IRQMSTAT) Field Descriptions

Bits	Field	Value	Description
31-2	Reserved	0	Always read as 0. Writes have no effect.
1	Bank1	0	No events pending from IRQ_STATUS_1
		1	At least one event is pending from IRQ_STATUS_1
0	Bank0	0	No events pending from IRQ_STATUS_0
		1	At least one event is pending from IRQ_STATUS_0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.2.5 USB1 IRQ_EOI Register (USB1IRQEOI)

The USB1 IRQ_EOI register (USB1IRQEOI) allows the CPU to acknowledge completion of an interrupt by writing to the end of interrupt (EOI). An eoi_write signal will be generated and another interrupt will be triggered if interrupt sources remain. This register will be reset one cycle after it has been written to.

The USB1 IRQ_EOI register is shown in Figure 15-91 and described in Table 15-101.

Figure 15-91. USB1 IRQ_EOI Register (USB1IRQEOI)

31	1	0
Reserved		EOI for USB1
R-0h		R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-101. USB1 IRQ_EOI Register (USB1IRQEOI) Field Descriptions

Bits	Field	Description
31-1	Reserved	Always read as 0. Writes have no effect.
0	EOI for USB1	EOI for USB1 interrupt

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.2.6 USB1 IRQ_STATUS_RAW_0 Register (USB1IRQSTATRAW0)

The USB1 IRQ_STATUS_RAW_0 register (USB1IRQSTATRAW0) allows the USB1 interrupt sources to be manually triggered when writing a 1 to a specific bit. A read of this register returns the USB1 interrupt event pending value.

General actions per bit:
Write 0: No action
Write 1: Set event
Read 0: No event pending
Read 1: Event pending

The USB1 IRQ_STATUS_RAW_0 register is shown in Figure 15-92 and described in Table 15-102.

Figure 15-92. USB1 IRQ_STATUS_RAW_0 Register (USB1IRQSTATRAW0)

31	30	29	28	27	26	25	24
RX EP 15	RX EP 14	RX EP 13	RX EP 12	RX EP 11	RX EP 10	RX EP 9	RX EP 8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RX EP 7	RX EP 6	RX EP 5	RX EP 4	RX EP 3	RX EP 2	RX EP 1	Reserved
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h
15	14	13	12	11	10	9	8
TX EP 15	TX EP 14	TX EP 13	TX EP 12	TX EP 11	TX EP 10	TX EP 9	TX EP 8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
TX EP 7	TX EP 6	TX EP 5	TX EP 4	TX EP 3	TX EP 2	TX EP 1	TX EP 0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-102. USB1 IRQ_STATUS_RAW_0 Register (USB1IRQSTATRAW1) Field Descriptions

Bits	Field	Description
31	RX EP 15	Interrupt status for RX endpoint 15
30	RX EP 14	Interrupt status for RX endpoint 14
29	RX EP 13	Interrupt status for RX endpoint 13
28	RX EP 12	Interrupt status for RX endpoint 12
27	RX EP 11	Interrupt status for RX endpoint 11
26	RX EP 10	Interrupt status for RX endpoint 10
25	RX EP 9	Interrupt status for RX endpoint 9
24	RX EP 8	Interrupt status for RX endpoint 8
23	RX EP 7	Interrupt status for RX endpoint 7
22	RX EP 6	Interrupt status for RX endpoint 6
21	RX EP 5	Interrupt status for RX endpoint 5
20	RX EP 4	Interrupt status for RX endpoint 4
19	RX EP 3	Interrupt status for RX endpoint 3
18	RX EP 2	Interrupt status for RX endpoint 2
17	RX EP 1	Interrupt status for RX endpoint 1
16	Reserved	Always read 0. Writes have no effect.
15	TX EP 15	Interrupt status for TX endpoint 15
14	TX EP 14	Interrupt status for TX endpoint 14
13	TX EP 13	Interrupt status for TX endpoint 13
12	TX EP 12	Interrupt status for TX endpoint 12
11	TX EP 11	Interrupt status for TX endpoint 11
10	TX EP 10	Interrupt status for TX endpoint 10

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-102. USB1 IRQ_STATUS_RAW_0 Register (USB1IRQSTATRAW1) Field Descriptions (continued)

Bits	Field	Description
9	TX EP 9	Interrupt status for TX endpoint 9
8	TX EP 8	Interrupt status for TX endpoint 8
7	TX EP 7	Interrupt status for TX endpoint 7
6	TX EP 6	Interrupt status for TX endpoint 6
5	TX EP 5	Interrupt status for TX endpoint 5
4	TX EP 4	Interrupt status for TX endpoint 4
3	TX EP 3	Interrupt status for TX endpoint 3
2	TX EP 2	Interrupt status for TX endpoint 2
1	TX EP 1	Interrupt status for TX endpoint 1
0	TX EP 0	Interrupt status for TX endpoint 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.2.7 USB1 IRQ_STATUS_0 Register (USB1IRQSTAT0)

The USB1 IRQ_STATUS_0 register (USB1IRQSTAT0) allows the USB1 interrupt sources to be manually cleared when writing a 1 to a specific bit. A read of this register returns the USB1 interrupt event pending value.

General actions per bit:
Write 0: No action
Write 1: Clear event
Read 0: No event pending
Read 1: Event pending

The USB1 IRQ_STATUS_0 register is shown in Figure 15-93 and described in Table 15-103.

Figure 15-93. USB1 IRQ_STATUS_0 Register (USB1IRQSTAT0)

31	30	29	28	27	26	25	24
RX EP 15	RX EP 14	RX EP 13	RX EP 12	RX EP 11	RX EP 10	RX EP 9	RX EP 8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RX EP 7	RX EP 6	RX EP 5	RX EP 4	RX EP 3	RX EP 2	RX EP 1	Reserved
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h
15	14	13	12	11	10	9	8
TX EP 15	TX EP 14	TX EP 13	TX EP 12	TX EP 11	TX EP 10	TX EP 9	TX EP 8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
TX EP 7	TX EP 6	TX EP 5	TX EP 4	TX EP 3	TX EP 2	TX EP 1	TX EP 0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-103. USB1 IRQ_STATUS_0 Register (USBIRQSTAT0) Field Descriptions

Bits	Field	Description
31	RX EP 15	Interrupt status for RX endpoint 15
30	RX EP 14	Interrupt status for RX endpoint 14
29	RX EP 13	Interrupt status for RX endpoint 13
28	RX EP 12	Interrupt status for RX endpoint 12
27	RX EP 11	Interrupt status for RX endpoint 11
26	RX EP 10	Interrupt status for RX endpoint 10
25	RX EP 9	Interrupt status for RX endpoint 9
24	RX EP 8	Interrupt status for RX endpoint 8
23	RX EP 7	Interrupt status for RX endpoint 7
22	RX EP 6	Interrupt status for RX endpoint 6
21	RX EP 5	Interrupt status for RX endpoint 5
20	RX EP 4	Interrupt status for RX endpoint 4
19	RX EP 3	Interrupt status for RX endpoint 3
18	RX EP 2	Interrupt status for RX endpoint 2
17	RX EP 1	Interrupt status for RX endpoint 1
16	Reserved	Always read 0. Writes have no effect.
15	TX EP 15	Interrupt status for TX endpoint 15
14	TX EP 14	Interrupt status for TX endpoint 14
13	TX EP 13	Interrupt status for TX endpoint 13
12	TX EP 12	Interrupt status for TX endpoint 12
11	TX EP 11	Interrupt status for TX endpoint 11
10	TX EP 10	Interrupt status for TX endpoint 10

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-103. USB1 IRQ_STATUS_0 Register (USBIRQSTAT0) Field Descriptions (continued)

Bits	Field	Description
9	TX EP 9	Interrupt status for TX endpoint 9
8	TX EP 8	Interrupt status for TX endpoint 8
7	TX EP 7	Interrupt status for TX endpoint 7
6	TX EP 6	Interrupt status for TX endpoint 6
5	TX EP 5	Interrupt status for TX endpoint 5
4	TX EP 4	Interrupt status for TX endpoint 4
3	TX EP 3	Interrupt status for TX endpoint 3
2	TX EP 2	Interrupt status for TX endpoint 2
1	TX EP 1	Interrupt status for TX endpoint 1
0	TX EP 0	Interrupt status for TX endpoint 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.2.8 USB1 IRQ_STATUS_1 Register (USB1IRQSTAT1)

The USB1 IRQ_STATUS_1 register (USB1IRQSTAT1) allows the USB1 interrupt sources to be manually cleared when writing a 1 to a specific bit. A read of this register returns the USB1 interrupt event pending value.

General actions per bit:
Write 0: No action
Write 1: Clear event
Read 0: No event pending
Read 1: Event pending

The USB1 IRQ_STATUS_1 register is shown in Figure 15-94 and described in Table 15-104.

Figure 15-94. USB1 IRQ_STATUS_1 Register (USB1IRQSTAT1)

31	30	29	28	27	26	25	24
TX FIFO 15	TX FIFO 14	TX FIFO 13	TX FIFO 12	TX FIFO 11	TX FIFO 10	TX FIFO 9	TX FIFO 8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
TX FIFO 7	TX FIFO 6	TX FIFO 5	TX FIFO 4	TX FIFO 3	TX FIFO 2	TX FIFO 1	Reserved
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h
15	14	13	12	11	10	9	8
Reserved						USB[9]	USB[9]
R-0h						R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
USB[7]	USB[6]	USB[5]	USB[4]	USB[3]	USB[2]	USB[1]	USB[0]
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-104. USB1 IRQ_STATUS_1 Register (USB1IRQSTAT1) Field Descriptions

Bits	Field	Description
31	TX FIFO 15	Interrupt status for TX FIFO endpoint 15
30	TX FIFO 14	Interrupt status for TX FIFO endpoint 14
29	TX FIFO 13	Interrupt status for TX FIFO endpoint 13
28	TX FIFO 12	Interrupt status for TX FIFO endpoint 12
27	TX FIFO 11	Interrupt status for TX FIFO endpoint 11
26	TX FIFO 10	Interrupt status for TX FIFO endpoint 10
25	TX FIFO 9	Interrupt status for TX FIFO endpoint 9
24	TX FIFO 8	Interrupt status for TX FIFO endpoint 8
23	TX FIFO 7	Interrupt status for TX FIFO endpoint 7
22	TX FIFO 6	Interrupt status for TX FIFO endpoint 6
21	TX FIFO 5	Interrupt status for TX FIFO endpoint 5
20	TX FIFO 4	Interrupt status for TX FIFO endpoint 4
19	TX FIFO 3	Interrupt status for TX FIFO endpoint 3
18	TX FIFO 2	Interrupt status for TX FIFO endpoint 2
17	TX FIFO 1	Interrupt status for TX FIFO endpoint 1
16-10	Reserved	Always read 0. Writes have no effect.
9	USB[9]	Interrupt status for Mentor controller USB_INT generic interrupt
8	USB[8]	Interrupt status for DRVVBUS level change
7	USB[7]	Interrupt status for VBUS < VBUS valid threshold
6	USB[6]	Interrupt status for SRP detected
5	USB[5]	Interrupt status for device disconnected (host mode)
4	USB[4]	Interrupt status for device connected (host mode)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-104. USB1 IRQ_STATUS_1 Register (USB1IRQSTAT1) Field Descriptions (continued)

Bits	Field	Description
3	USB[3]	Interrupt status for SOF started
2	USB[2]	Interrupt status for Reset signaling detected (peripheral mode) Babble detected (host mode)
1	USB[1]	Interrupt status for Resume signaling detected
0	USB[0]	Interrupt status for Suspend signaling detected

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.2.9 USB1 IRQ_ENABLE_SET_0 Register (USB1IRQENABLESET0)

The USB1 IRQ_ENABLE_SET_0 register (USB1IRQENABLESET0) allows the USB1 interrupt sources to be manually enabled when writing a 1 to a specific bit. A read of this register returns the USB1 interrupt enabled value.

General actions per bit:
Write 0: No action
Write 1: Interrupt enabled
Read 0: Interrupt disabled
Read 1: Interrupt enabled

The USB1 IRQ_ENABLE_SET_0 register is shown in Figure 15-95 and described in Table 15-105.

Figure 15-95. USB1 IRQ_ENABLE_SET_0 Register (USB1IRQENABLESET0)

31	30	29	28	27	26	25	24
RX EP 15	RX EP 14	RX EP 13	RX EP 12	RX EP 11	RX EP 10	RX EP 9	RX EP 8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RX EP 7	RX EP 6	RX EP 5	RX EP 4	RX EP 3	RX EP 2	RX EP 1	Reserved
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h
15	14	13	12	11	10	9	8
TX EP 15	TX EP 14	TX EP 13	TX EP 12	TX EP 11	TX EP 10	TX EP 9	TX EP 8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
TX EP 7	TX EP 6	TX EP 5	TX EP 4	TX EP 3	TX EP 2	TX EP 1	TX EP 0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-105. USB1 IRQ_ENABLE_SET_0 Register (USB1IRQENABLESET0) Field Descriptions

Bits	Field	Description
31	RX EP 15	Interrupt enable for RX endpoint 15
30	RX EP 14	Interrupt enable for RX endpoint 14
29	RX EP 13	Interrupt enable for RX endpoint 13
28	RX EP 12	Interrupt enable for RX endpoint 12
27	RX EP 11	Interrupt enable for RX endpoint 11
26	RX EP 10	Interrupt enable for RX endpoint 10
25	RX EP 9	Interrupt enable for RX endpoint 9
24	RX EP 8	Interrupt enable for RX endpoint 8
23	RX EP 7	Interrupt enable for RX endpoint 7
22	RX EP 6	Interrupt enable for RX endpoint 6
21	RX EP 5	Interrupt enable for RX endpoint 5
20	RX EP 4	Interrupt enable for RX endpoint 4
19	RX EP 3	Interrupt enable for RX endpoint 3
18	RX EP 2	Interrupt enable for RX endpoint 2
17	RX EP 1	Interrupt enable for RX endpoint 1
16	Reserved	Always read 0. Writes have no effect.
15	TX EP 15	Interrupt enable for TX endpoint 15
14	TX EP 14	Interrupt enable for TX endpoint 14
13	TX EP 13	Interrupt enable for TX endpoint 13
12	TX EP 12	Interrupt enable for TX endpoint 12
11	TX EP 11	Interrupt enable for TX endpoint 11
10	TX EP 10	Interrupt enable for TX endpoint 10

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-105. USB1 IRQ_ENABLE_SET_0 Register (USB1IRQENABLESET0) Field Descriptions (continued)

Bits	Field	Description
9	TX EP 9	Interrupt enable for TX endpoint 9
8	TX EP 8	Interrupt enable for TX endpoint 8
7	TX EP 7	Interrupt enable for TX endpoint 7
6	TX EP 6	Interrupt enable for TX endpoint 6
5	TX EP 5	Interrupt enable for TX endpoint 5
4	TX EP 4	Interrupt enable for TX endpoint 4
3	TX EP 3	Interrupt enable for TX endpoint 3
2	TX EP 2	Interrupt enable for TX endpoint 2
1	TX EP 1	Interrupt enable for TX endpoint 1
0	TX EP 0	Interrupt enable for TX endpoint 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.2.10 USB1 IRQ_ENABLE_SET_1 Register (USB1IRQENABLESET1)

The USB1 IRQ_ENABLE_SET_1 register (USB1IRQENABLESET1) allows the USB1 interrupt sources to be manually enabled when writing a 1 to a specific bit. A read of this register returns the USB1 interrupt enabled value.

General actions per bit:
Write 0: No action
Write 1: Interrupt enabled
Read 0: Interrupt disabled
Read 1: Interrupt enabled

The USB1 IRQ_ENABLE_SET_1 register is shown in Figure 15-96 and described in Table 15-106.

Figure 15-96. USB1 IRQ_ENABLE_SET_1 Register (USB1IRQENABLESET1)

31	30	29	28	27	26	25	24
TX FIFO 15	TX FIFO 14	TX FIFO 13	TX FIFO 12	TX FIFO 11	TX FIFO 10	TX FIFO 9	TX FIFO 8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
TX FIFO 15	TX FIFO 15	TX FIFO 15	TX FIFO 15	TX FIFO 15	TX FIFO 15	TX FIFO 15	Reserved
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h
15	14	13	12	11	10	9	8
Reserved						USB[9]	USB[9]
R-0h						R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
USB[7]	USB[6]	USB[5]	USB[4]	USB[3]	USB[2]	USB[1]	USB[0]
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-106. USB1 IRQ_ENABLE_SET_1 Register (USB1IRQENABLESET1) Field Descriptions

Bits	Field	Description
31	TX FIFO 15	Interrupt enable for TX FIFO endpoint 15
30	TX FIFO 14	Interrupt enable for TX FIFO endpoint 14
29	TX FIFO 13	Interrupt enable for TX FIFO endpoint 13
28	TX FIFO 12	Interrupt enable for TX FIFO endpoint 12
27	TX FIFO 11	Interrupt enable for TX FIFO endpoint 11
26	TX FIFO 10	Interrupt enable for TX FIFO endpoint 10
25	TX FIFO 9	Interrupt enable for TX FIFO endpoint 9
24	TX FIFO 8	Interrupt enable for TX FIFO endpoint 8
23	TX FIFO 7	Interrupt enable for TX FIFO endpoint 7
22	TX FIFO 6	Interrupt enable for TX FIFO endpoint 6
21	TX FIFO 5	Interrupt enable for TX FIFO endpoint 5
20	TX FIFO 4	Interrupt enable for TX FIFO endpoint 4
19	TX FIFO 3	Interrupt enable for TX FIFO endpoint 3
18	TX FIFO 2	Interrupt enable for TX FIFO endpoint 2
17	TX FIFO 1	Interrupt enable for TX FIFO endpoint 1
16-10	Reserved	Always read 0. Writes have no effect.
9	USB[9]	Interrupt enable for Mentor controller USB_INT generic interrupt
8	USB[8]	Interrupt enable for DRVVBUS level change
7	USB[7]	Interrupt enable for VBUS < VBUS valid threshold
6	USB[6]	Interrupt enable for SRP detected
5	USB[5]	Interrupt enable for device disconnected (host mode)
4	USB[4]	Interrupt enable for device connected (host mode)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-106. USB1 IRQ_ENABLE_SET_1 Register (USB1IRQENABLESET1) Field Descriptions (continued)

Bits	Field	Description
3	USB[3]	Interrupt enable for SOF started
2	USB[2]	Interrupt enable for Reset signaling detected (peripheral mode) Babble detected (host mode)
1	USB[1]	Interrupt enable for Resume signaling detected
0	USB[0]	Interrupt enable for Suspend signaling detected

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.2.11 USB1 IRQ_ENABLE_CLR_0 Register (USB1IRQENABLECLR0)

The USB1 IRQ_ENABLE_CLR_0 register (USB1IRQENABLECLR0) allows the USB1 interrupt sources to be manually disabled when writing a 1 to a specific bit. A read of this register returns the USB1 interrupt enabled value.

General actions per bit:
Write 0: No action
Write 1: Disable interrupt
Read 0: Interrupt disabled
Read 1: Interrupt enabled

The USB1 IRQ_ENABLE_CLR_0 register is shown in Figure 15-97 and described in Table 15-107.

Figure 15-97. USB1 IRQ_ENABLE_CLR_0 Register (USB1IRQENABLECLR0)

31	30	29	28	27	26	25	24
RX EP 15	RX EP 14	RX EP 13	RX EP 12	RX EP 11	RX EP 10	RX EP 9	RX EP 8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
RX EP 7	RX EP 6	RX EP 5	RX EP 4	RX EP 3	RX EP 2	RX EP 1	Reserved
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h
15	14	13	12	11	10	9	8
TX EP 15	TX EP 14	TX EP 13	TX EP 12	TX EP 11	TX EP 10	TX EP 9	TX EP 8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
TX EP 7	TX EP 6	TX EP 5	TX EP 4	TX EP 3	TX EP 2	TX EP 1	TX EP 0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-107. USB1 IRQ_ENABLE_CLR_0 Register (USB1IRQENABLECLR0) Field Descriptions

Bits	Field	Description
31	RX EP 15	Interrupt enable for RX endpoint 15
30	RX EP 14	Interrupt enable for RX endpoint 14
29	RX EP 13	Interrupt enable for RX endpoint 13
28	RX EP 12	Interrupt enable for RX endpoint 12
27	RX EP 11	Interrupt enable for RX endpoint 11
26	RX EP 10	Interrupt enable for RX endpoint 10
25	RX EP 9	Interrupt enable for RX endpoint 9
24	RX EP 8	Interrupt enable for RX endpoint 8
23	RX EP 7	Interrupt enable for RX endpoint 7
22	RX EP 6	Interrupt enable for RX endpoint 6
21	RX EP 5	Interrupt enable for RX endpoint 5
20	RX EP 4	Interrupt enable for RX endpoint 4
19	RX EP 3	Interrupt enable for RX endpoint 3
18	RX EP 2	Interrupt enable for RX endpoint 2
17	RX EP 1	Interrupt enable for RX endpoint 1
16	Reserved	Always read 0. Writes have no effect.
15	TX EP 15	Interrupt enable for TX endpoint 15
14	TX EP 14	Interrupt enable for TX endpoint 14
13	TX EP 13	Interrupt enable for TX endpoint 13
12	TX EP 12	Interrupt enable for TX endpoint 12
11	TX EP 11	Interrupt enable for TX endpoint 11
10	TX EP 10	Interrupt enable for TX endpoint 10

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-107. USB1 IRQ_ENABLE_CLR_0 Register (USB1IRQENABLECLR0) Field Descriptions (continued)

Bits	Field	Description
9	TX EP 9	Interrupt enable for TX endpoint 9
8	TX EP 8	Interrupt enable for TX endpoint 8
7	TX EP 7	Interrupt enable for TX endpoint 7
6	TX EP 6	Interrupt enable for TX endpoint 6
5	TX EP 5	Interrupt enable for TX endpoint 5
4	TX EP 4	Interrupt enable for TX endpoint 4
3	TX EP 3	Interrupt enable for TX endpoint 3
2	TX EP 2	Interrupt enable for TX endpoint 2
1	TX EP 1	Interrupt enable for TX endpoint 1
0	TX EP 0	Interrupt enable for TX endpoint 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.2.12 USB1 IRQ_ENABLE_CLR_1 Register (USB1IREENABLECLR1)

The USB1 IRQ_ENABLE_CLR_1 register (USB1IREENABLECLR1) allows the USB1 interrupt sources to be manually disabled when writing a 1 to a specific bit. A read of this register returns the USB1 interrupt enabled value.

General actions per bit:
Write 0: No action
Write 1: Disable interrupt
Read 0: Interrupt disabled
Read 1: Interrupt enabled

The USB1 IRQ_ENABLE_CLR_1 register is shown in Figure 15-98 and described in Table 15-108.

Figure 15-98. USB1 IRQ_ENABLE_CLR_1 Register (USB1IREENABLECLR1)

31	30	29	28	27	26	25	24
TX FIFO 15	TX FIFO 14	TX FIFO 13	TX FIFO 12	TX FIFO 11	TX FIFO 10	TX FIFO 9	TX FIFO 8
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
23	22	21	20	19	18	17	16
TX FIFO 15	TX FIFO 15	TX FIFO 15	TX FIFO 15	TX FIFO 15	TX FIFO 15	TX FIFO 15	Reserved
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h
15	14	13	12	11	10	9	8
Reserved						USB[9]	USB[9]
R-0h						R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
USB[7]	USB[6]	USB[5]	USB[4]	USB[3]	USB[2]	USB[1]	USB[0]
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-108. USB1 IRQ_ENABLE_CLR_1 Register (USB1IREENABLECLR1) Field Descriptions

Bits	Field	Description
31	TX FIFO 15	Interrupt enable for TX FIFO endpoint 15
30	TX FIFO 14	Interrupt enable for TX FIFO endpoint 14
29	TX FIFO 13	Interrupt enable for TX FIFO endpoint 13
28	TX FIFO 12	Interrupt enable for TX FIFO endpoint 12
27	TX FIFO 11	Interrupt enable for TX FIFO endpoint 11
26	TX FIFO 10	Interrupt enable for TX FIFO endpoint 10
25	TX FIFO 9	Interrupt enable for TX FIFO endpoint 9
24	TX FIFO 8	Interrupt enable for TX FIFO endpoint 8
23	TX FIFO 7	Interrupt enable for TX FIFO endpoint 7
22	TX FIFO 6	Interrupt enable for TX FIFO endpoint 6
21	TX FIFO 5	Interrupt enable for TX FIFO endpoint 5
20	TX FIFO 4	Interrupt enable for TX FIFO endpoint 4
19	TX FIFO 3	Interrupt enable for TX FIFO endpoint 3
18	TX FIFO 2	Interrupt enable for TX FIFO endpoint 2
17	TX FIFO 1	Interrupt enable for TX FIFO endpoint 1
16-10	Reserved	Always read 0. Writes have no effect.
9	USB[9]	Interrupt enable for Mentor controller USB_INT generic interrupt
8	USB[8]	Interrupt enable for DRVVBUS level change
7	USB[7]	Interrupt enable for VBUS < VBUS valid threshold
6	USB[6]	Interrupt enable for SRP detected
5	USB[5]	Interrupt enable for device disconnected (host mode)
4	USB[4]	Interrupt enable for device connected (host mode)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-108. USB1 IRQ_ENABLE_CLR_1 Register (USB1IREENABLECLR1) Field Descriptions (continued)

Bits	Field	Description
3	USB[3]	Interrupt enable for SOF started
2	USB[2]	Interrupt enable for Reset signaling detected (peripheral mode) Babble detected (host mode)
1	USB[1]	Interrupt enable for Resume signaling detected
0	USB[0]	Interrupt enable for Suspend signaling detected

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.2.13 USB1 Tx Mode Register (USB1TXMODE)

The USB1 Tx mode register (USB1TXMODE) allows the CPU to individually enable RNDIS/Generic/CDC modes for each endpoint. Using the global RNDIS enable in the Control Register overrides this register and enable RNDIS mode for all endpoints.

The USB1 Tx mode register is shown in Figure 15-99 and described in Table 15-109.

Figure 15-99. USB1 Tx Mode Register (USB1TXMODE)

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
Reserved	Tx15_mode	Tx14_mode	Tx13_mode	Tx12_mode	Tx11_mode	Tx10_mode	Tx9_mode
R/0	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
Tx8_mode	Tx7_mode	Tx6_mode	Tx5_mode	Tx4_mode	Tx3_mode	Tx2_mode	Tx1_mode
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-109. USB1 Tx Mode Register (USB1TXMODE) Field Descriptions

Bits	Field	Value	Description
31-30	Reserved	0	Always read as 0. Writes have no effect.
29-28	Tx15_mode		
		00	Transparent Mode on TX endpoint 15
		01	RNDIS MODE on TX endpoint 15
		10	CDC Mode on TX endpoint 15
		11	Generic RNDIS Mode on TX endpoint 15
27-26	Tx14_mode		
		00	Transparent Mode on TX endpoint 14
		01	RNDIS MODE on TX endpoint 14
		10	CDC Mode on TX endpoint 14
		11	Generic RNDIS Mode on TX endpoint 14
25-24	Tx13_mode		
		00	Transparent Mode on TX endpoint 13
		01	RNDIS MODE on TX endpoint 13
		10	CDC Mode on TX endpoint 13
		11	Generic RNDIS Mode on TX endpoint 13
23-22	Tx12_mode		
		00	Transparent Mode on TX endpoint 12
		01	RNDIS MODE on TX endpoint 12
		10	CDC Mode on TX endpoint 12
		11	Generic RNDIS Mode on TX endpoint 12
21-20	Tx11_mode		
		00	Transparent Mode on TX endpoint 11
		01	RNDIS MODE on TX endpoint 11
		10	CDC Mode on TX endpoint 11
		11	Generic RNDIS Mode on TX endpoint 11
19-18	Tx10_mode		
		00	Transparent Mode on TX endpoint 10
		01	RNDIS MODE on TX endpoint 10
		10	CDC Mode on TX endpoint 10
		11	Generic RNDIS Mode on TX endpoint 10

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-109. USB1 Tx Mode Register (USB1TXMODE) Field Descriptions (continued)

Bits	Field	Value	Description
17-16	Tx9_mode		
		00	Transparent Mode on TX endpoint 9
		01	RNDIS MODE on TX endpoint 9
		10	CDC Mode on TX endpoint 9
		11	Generic RNDIS Mode on TX endpoint 9
15-14	Tx8_mode		
		00	Transparent Mode on TX endpoint 8
		01	RNDIS MODE on TX endpoint 8
		10	CDC Mode on TX endpoint 8
		11	Generic RNDIS Mode on TX endpoint 8
13-12	Tx7_mode		
		00	Transparent Mode on TX endpoint 7
		01	RNDIS MODE on TX endpoint 7
		10	CDC Mode on TX endpoint 7
		11	Generic RNDIS Mode on TX endpoint 7
11-10	Tx6_mode		
		00	Transparent Mode on TX endpoint 6
		01	RNDIS MODE on TX endpoint 6
		10	CDC Mode on TX endpoint 6
		11	Generic RNDIS Mode on TX endpoint 6
9-8	Tx5_mode		
		00	Transparent Mode on TX endpoint 5
		01	RNDIS MODE on TX endpoint 5
		10	CDC Mode on TX endpoint 5
		11	Generic RNDIS Mode on TX endpoint 5
7-6	Tx4_mode		
		00	Transparent Mode on TX endpoint 4
		01	RNDIS MODE on TX endpoint 4
		10	CDC Mode on TX endpoint 4
		11	Generic RNDIS Mode on TX endpoint 4
5-4	Tx3_mode		
		00	Transparent Mode on TX endpoint 3
		01	RNDIS MODE on TX endpoint 3
		10	CDC Mode on TX endpoint 3
		11	Generic RNDIS Mode on TX endpoint 3
3-2	Tx2_mode		
		00	Transparent Mode on TX endpoint 2
		01	RNDIS MODE on TX endpoint 2
		10	CDC Mode on TX endpoint 2
		11	Generic RNDIS Mode on TX endpoint 2
1-0	Tx1_mode		
		00	Transparent Mode on TX endpoint 1
		01	RNDIS MODE on TX endpoint 1
		10	CDC Mode on TX endpoint 1
		11	Generic RNDIS Mode on TX endpoint 1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.2.14 USB1 Rx Mode Register (USB1RXMODE)

The USB1 Rx mode register (USB1RXMODE) allows the CPU to individually enable RNDIS/Generic/CDC modes for each endpoint. Using the global RNDIS enable in the control register overrides this register and enable RNDIS mode for all endpoints.

The USB1 Rx mode register is shown in Figure 15-100 and described in Table 15-110.

Figure 15-100. USB1 Rx Mode Register (USB1RXMODE)

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16
Reserved	Rx15_mode	Rx14_mode	Rx13_mode	Rx12_mode	Rx11_mode	Rx10_mode	Rx9_mode
R/0	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
Rx8_mode	Rx7_mode	Rx6_mode	Rx5_mode	Rx4_mode	Rx3_mode	Rx2_mode	Rx1_mode
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-110. USB1 Rx Mode Register (USB1RXMODE) Field Descriptions

Bits	Field	Value	Description
31-30	Reserved	0	Always read as 0. Writes have no effect.
29-28	Rx15_mode		
		00	Transparent Mode on RX endpoint 15
		01	RNDIS MODE on RX endpoint 15
		10	CDC Mode on RX endpoint 15
		11	Generic RNDIS Mode on RX endpoint 15
27-26	Rx14_mode		
		00	Transparent Mode on RX endpoint 14
		01	RNDIS MODE on RX endpoint 14
		10	CDC Mode on RX endpoint 14
		11	Generic RNDIS Mode on RX endpoint 14
25-24	Rx13_mode		
		00	Transparent Mode on RX endpoint 13
		01	RNDIS MODE on RX endpoint 13
		10	CDC Mode on RX endpoint 13
		11	Generic RNDIS Mode on RX endpoint 13
23-22	Rx12_mode		
		00	Transparent Mode on RX endpoint 12
		01	RNDIS MODE on RX endpoint 12
		10	CDC Mode on RX endpoint 12
		11	Generic RNDIS Mode on RX endpoint 12
21-20	Rx11_mode		
		00	Transparent Mode on RX endpoint 11
		01	RNDIS MODE on RX endpoint 11
		10	CDC Mode on RX endpoint 11
		11	Generic RNDIS Mode on RX endpoint 11
19-18	Rx10_mode		
		00	Transparent Mode on RX endpoint 10
		01	RNDIS MODE on RX endpoint 10
		10	CDC Mode on RX endpoint 10
		11	Generic RNDIS Mode on RX endpoint 10

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-110. USB1 Rx Mode Register (USB1RXMODE) Field Descriptions (continued)

Bits	Field	Value	Description
17-16	Rx9_mode		
		00	Transparent Mode on RX endpoint 9
		01	RNDIS MODE on RX endpoint 9
		10	CDC Mode on RX endpoint 9
		11	Generic RNDIS Mode on RX endpoint 9
15-14	Rx8_mode		
		00	Transparent Mode on RX endpoint 8
		01	RNDIS MODE on RX endpoint 8
		10	CDC Mode on RX endpoint 8
		11	Generic RNDIS Mode on RX endpoint 8
13-12	Rx7_mode		
		00	Transparent Mode on RX endpoint 7
		01	RNDIS MODE on RX endpoint 7
		10	CDC Mode on RX endpoint 7
		11	Generic RNDIS Mode on RX endpoint 7
11-10	Rx6_mode		
		00	Transparent Mode on RX endpoint 6
		01	RNDIS MODE on RX endpoint 6
		10	CDC Mode on RX endpoint 6
		11	Generic RNDIS Mode on RX endpoint 6
9-8	Rx5_mode		
		00	Transparent Mode on RX endpoint 5
		01	RNDIS MODE on RX endpoint 5
		10	CDC Mode on RX endpoint 5
		11	Generic RNDIS Mode on RX endpoint 5
7-6	Rx4_mode		
		00	Transparent Mode on RX endpoint 4
		01	RNDIS MODE on RX endpoint 4
		10	CDC Mode on RX endpoint 4
		11	Generic RNDIS Mode on RX endpoint 4
5-4	Rx3_mode		
		00	Transparent Mode on RX endpoint 3
		01	RNDIS MODE on RX endpoint 3
		10	CDC Mode on RX endpoint 3
		11	Generic RNDIS Mode on RX endpoint 3
3-2	Rx2_mode		
		00	Transparent Mode on RX endpoint 2
		01	RNDIS MODE on RX endpoint 2
		10	CDC Mode on RX endpoint 2
		11	Generic RNDIS Mode on RX endpoint 2
1-0	Rx1_mode		
		00	Transparent Mode on RX endpoint 1
		01	RNDIS MODE on RX endpoint 1
		10	CDC Mode on RX endpoint 1
		11	Generic RNDIS Mode on RX endpoint 1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.2.15 USB1 Generic RNDIS EP N Size Register (USB1GENRNDISEPn)

The USB1 generic RNDIS EP N size register (USB1GENRNDISEPn) is programmed with a RNDIS packet size in bytes. When EP(N) is in Generic RNDIS mode, the received USB packets will be collected into a single CPPI packet that is completed when the number of bytes equal to the value of this register has been received, or a “short” packet is received. Note that this register must be programmed with a value that is an integer multiple of the endpoint size. The maximum value this register can be programmed with is 10000h, or 65536. N can range from 1 to 15.

The USB1 generic RNDIS EP N size register is shown in Figure 15-101 and described in Table 15-111.

Figure 15-101. USB1 Generic RNDIS EP N Size Register (USB1GENRNDISEPn)

31 – 17	16 – 0
Reserved	Ep(N)_size
R-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-111. USB1 Generic RNDIS EP N Size Register (USB1GENRNDISEPn) Field Descriptions

Bits	Field	Description
31-17	Reserved	Always read as 0. Writes have no effect.
16-0	Ep(N)_size	Generic RNDIS packet size.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.2.16 USB1 Auto Req Register (USB1AUTOREQ)

The USB1 auto req register (USB1AUTOREQ) allows the CPU to enable an automatic IN token request generation for host mode RX operation per each RX endpoint. This features has the DMA set the ReqPkt bit in the RXCSR when it clears the RxPktRdy bit after reading out a packet. The ReqPkt bit is used by the core to generate an IN token to receive data. By using this feature, the host can automatically generate an IN token after the DMA finishes receiving data and empties an endpoint buffer, thus receiving the next data packet as soon as possible from the connected device. Without this feature, the CPU will have to manually set the ReqPkt bit for every USB packet.

There are two modes that Auto Req can function: always or all except an EOP. The always mode will set the ReqPkt bit after every USB packet the DMA receives thus generating a new IN token after each USB packet. The EOP mode will set the ReqPkt bit after every USB packet that isn't an EOP (end of packet) in the CPPI descriptor. For RNDIS, CDC, and Generic RNDIS modes the auto req will stop when the EOP is received (either via a short packet for RNDIS,CDC, and Generic RNDIS or the count is reached for Generic RNDIS), making it useful for starting a large RNDIS packet and having it auto generate IN tokens until the end of the RNDIS packet. For transparent mode, every USB packet is an EOP CPPI packet, so the auto req never functions and acts like auto req is disabled.

The USB1 auto req register is shown in Figure 15-102 and described in Table 15-112.

Figure 15-102. USB1 Auto Req Register (USB1AUTOREQ)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		Rx15_autoreq		Rx14_autoreq		Rx13_autoreq		Rx12_autoreq		Rx11_autoreq		Rx10_autoreq		Rx9_autoreq	
R-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rx8_autoreq		Rx7_autoreq		Rx6_autoreq		Rx5_autoreq		Rx4_autoreq		Rx3_autoreq		Rx2_autoreq		Rx1_autoreq	
R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h		R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-112. USB1 Auto Req Register (USB1AUTOREQ) Field Descriptions

Bits	Field	Values	Description
31-30	Reserved	0	Always read as 0. Writes have no effect.
29-28	Rx15_autoreq		RX endpoint 15 Auto Req enable
		00	No auto req
		01	Auto req on all but EOP
		10	Reserved
		11	Auto req always
27-26	Rx14_autoreq		RX endpoint 14 Auto Req enable
		00	No auto req
		01	Auto req on all but EOP
		10	Reserved
		11	Auto req always
25-24	Rx13_autoreq		RX endpoint 13 Auto Req enable
		00	No auto req
		01	Auto req on all but EOP
		10	Reserved
		11	Auto req always
23-22	Rx12_autoreq		RX endpoint 12 Auto Req enable
		00	No auto req
		01	Auto req on all but EOP
		10	Reserved
		11	Auto req always

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-112. USB1 Auto Req Register (USB1AUTOREQ) Field Descriptions (continued)

Bits	Field	Values	Description
21-20	Rx11_autoreq		RX endpoint 11 Auto Req enable
		00	No auto req
		01	Auto req on all but EOP
		10	Reserved
		11	Auto req always
19-18	Rx10_autoreq		RX endpoint 10 Auto Req enable
		00	No auto req
		01	Auto req on all but EOP
		10	Reserved
		11	Auto req always
17-16	Rx9_autoreq		RX endpoint 9 Auto Req enable
		00	No auto req
		01	Auto req on all but EOP
		10	Reserved
		11	Auto req always
15-14	Rx8_autoreq		RX endpoint 8 Auto Req enable
		00	No auto req
		01	Auto req on all but EOP
		10	Reserved
		11	Auto req always
13-12	Rx7_autoreq		RX endpoint 7 Auto Req enable
		00	No auto req
		01	Auto req on all but EOP
		10	Reserved
		11	Auto req always
11-10	Rx6_autoreq		RX endpoint 6 Auto Req enable
		00	No auto req
		01	Auto req on all but EOP
		10	Reserved
		11	Auto req always
9-8	Rx5_autoreq		RX endpoint 5 Auto Req enable
		00	No auto req
		01	Auto req on all but EOP
		10	Reserved
		11	Auto req always
7-6	Rx4_autoreq		RX endpoint 4 Auto Req enable
		00	No auto req
		01	Auto req on all but EOP
		10	Reserved
		11	Auto req always
5-4	Rx3_autoreq		RX endpoint 3 Auto Req enable
		00	No auto req
		01	Auto req on all but EOP
		10	Reserved
		11	Auto req always

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-112. USB1 Auto Req Register (USB1AUTOREQ) Field Descriptions (continued)

Bits	Field	Values	Description
3-2	Rx2_autoreq		RX endpoint 2 Auto Req enable
		00	No auto req
		01	Auto req on all but EOP
		10	Reserved
		11	Auto req always
1-0	Rx1_autoreq		RX endpoint 1 Auto Req enable
		00	No auto req
		01	Auto req on all but EOP
		10	Reserved
		11	Auto req always

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.2.17 USB1 SRP Fix Time Register (USB1SRPFIXTIME)

The USB1 SRP fix time register (USB1SRPFIXTIME) allows the CPU to configure the maximum amount of time the SRP fix logic blocks the AVAID from the PHY to the OTG core. This time allows the VBUS signal the ability to get below the thresholds and therefore remove the chance of voltage bounces which could give false threshold measurements.

The USB1 SRP fix time register is shown in Figure 15-103 and described in Table 15-113.

Figure 15-103. USB1 SRP Fix Time Register (USB1SRPFIXTIME)

31	0
srpfixtime	
R/W-0280de80h	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-113. USB1 SRP Fix Time Register (USB1SRPFIXTIME) Field Descriptions

Bits	Field	Description
31-0	srpfixtime	SRP Fix maximum time in 60 MHz cycles. Default is 700 ms.

15.5.2.2.18 USB1 Teardown Register (USB1TDOWN)

The USB1 teardown register (USB1TDOWN) controls the tearing down of rx and tx fifos in the USB controller. When a '1' is written to a valid bit in this register, the CPPI FIFO pointers for that endpoint are cleared , and the register automatically clears itself after 1 clock cycle. This register must be used in conjunction with the CPPI DMA Teardown mechanism. The Host should also write the FlushFIFO bits in the TXCSR and RXCSR Mentor USB Controller registers to ensure a complete teardown of the endpoint. See the Mentor specification for details.

The USB1 teardown register is shown in Figure 15-104 and described in Table 15-114.

Figure 15-104. USB1 Teardown Register (USB1TDOWN)

31	17	16	15	0
tx_tdown		Rsvd	rx_tdown	Rsvd
R/W-0h		R-0h	R/W-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-114. USB1 Teardown Register (USB1TDOWN) Field Descriptions

Bits	Field	Description
31-17	tx_tdown	Tx Endpoint Teardown. Write '1' to corresponding bit to set. Read as '0'. Bit 31 = Endpoint 15 ... Bit 17 = Endpoint 1
16	Reserved	Always read as 0. Writes have no effect.
15-1	rx_tdown	RX Endpoint Teardown. Write '1' to corresponding bit to set. Read as '0'. Bit 15 = Endpoint 15 ... Bit 1 = Endpoint 1
0	Reserved	Always read as 0. Writes have no effect.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.2.19 USB1 PHY UTMI Register (USB1UTMI)

The USB1 PHY UTMI register (USB1UTMI) controls various PHY UTMI signals. The UTMI interface is located between the PHY module and the Mentor Graphics Controller. These signals are inputs to the PHY but are not outputs from the Mentor Graphics Controller. It is important to set the otgdisable bit in to a high value to operate the PHY in a non-OTG mode.

The USB1 PHY UTMI register is shown in Figure 15-105 and described in Table 15-115.

Figure 15-105. USB1 PHY UTMI Register (USB1UTMI)

31							24
Reserved							
R-0h							

23	22	21	20	19	18	17	16
txbitstuffen	txbitstuffenh	otgdisable	vbusvldextsel	vbusvldext	txenablen	fsxcvrowner	txvalidh
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

15							8
datainh							
R/W-0h							

7					3	2	1	0
Reserved						wordinterface	fsdataext	fsse0ext
R-0h						R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-115. USB1 PHY UTMI Register (USB1UTMI) Field Descriptions

Bits	Field	Description
31-24	Reserved	Always read as 0. Writes have no effect.
23	txbitstuffen	PHY UTMI input for signal txbitstuffen
22	txbitstuffenh	PHY UTMI input for signal txbitstuffenh
21	otgdisable	PHY UTMI input for signal otgdisable
20	vbusvldextsel	PHY UTMI input for signal vbusvldextsel
19	vbusvldext	PHY UTMI input for signal vbusvldext
18	txenablen	PHY UTMI input for signal txenablen
17	fsxcvrowner	PHY UTMI input for signal fsxcvrowner
16	txvalidh	PHY UTMI input for signal txvalidh
15-8	datainh	PHY UTMI input for signal datainh
7-3	Reserved	Always read as 0. Writes have no effect.
2	wordinterface	PHY UTMI input for signal wordinterface
1	fsdataext	PHY UTMI input for signal fsdataext
0	fsse0ext	PHY UTMI input for signal fsse0ext

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.2.20 USB1 MGC UTMI Loopback Register (USB1UTMILB)

The USB1 MGC UTMI loopback register (USB1UTMILB) contains most of the input and output UTMI signals for the Mentor Graphics Controller. The UTMI interface is located between the PHY module and the Mentor Graphics Controller.

In the loopback mode test register bits 11 to 0 control various UTMI signals that are input to the Mentor Graphics Controller.

In the loopback mode test register bits 28 to 16 observe various UTMI signals that are output from the Mentor Graphics Controller.

The USB1 MGC UTMI loopback register is shown in Figure 15-106 and described in Table 15-116.

Figure 15-106. USB1 MGC UTMI Loopback Register (USB1UTMILB)

31		29	28	27	26	25	24
Reserved			suspendm	opmode		txvalid	xcvrsel
R-0h			R-0h	R-0h		R-0h	R-0h

23	22	21	20	19	18	17	16
xcvrsel	termsel	drvvbus	chrgvbus	dischrgvbus	dppulldown	dmpulldown	idpullup
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

15			12	11	10	9	8
Reserved				iddig	hostdiscon	sessend	avalid
				R/W-0h	R/W-0h	R/W-0h	R/W-0h

7	6	5	4	3	2	1	0
vbusvalid	rxerror	Reserved		linestate		Reserved	
R/W-0h	R/W-0h	R-0h		R/W-0h		R-0h	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-116. USB1 MGC UTMI Loopback Register (USB1UTMILB) Field Descriptions

Bits	Field	Description
31-29	Reserved	Always read as 0. Writes have no effect.
28	suspendm	Loopback test observed value for suspendm
27-26	opmode	Loopback test observed value for opmode
25	txvalid	Loopback test observed value for txvalid
24-23	xcvrsel	Loopback test observed value for xcvrsel
22	termsel	Loopback test observed value for termsel
21	drvvbus	Loopback test observed value for drvvbus
20	chrgvbus	Loopback test observed value for chrgvbus
19	dischrgvbus	Loopback test observed value for dischrgvbus
18	dppulldown	Loopback test observed value for dppulldown
17	dmpulldown	Loopback test observed value for dmpulldown
16	idpullup	Loopback test observed value for idpullup
15-12	Reserved	Always read as 0. Writes have no effect.
11	iddig	Loopback test value for iddig
10	hostdiscon	Loopback test value for hostdiscon
9	sessend	Loopback test value for sessend
8	avalid	Loopback test value for avalid
7	vbusvalid	Loopback test value for vbusvalid
6	rxerror	Loopback test value for rxerror
5-4	Reserved	Always read as 0. Writes have no effect.
3-2	linestate	Loopback test value for linestate
1-0	Reserved	Always read as 0. Writes have no effect.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.2.2.21 USB1 Mode Register (USB1MODE)

The USB1 mode register (USB1MODE) allows the user a flexibility of controlling the role that the controller assumes. Via the setting of the USB1MODE[Bit7=Iddig_Mux], the user has the capability of allowing the control to be coming from a register field (USB1MODE[Bit8=iddig]) or from the USB_ID pin which is indirectly controlled by the cable end. if USB1MODE[iddig_mux]=0, the state of the USB_ID pin controls the role the controller assumes. However, if USB1MODE[Iddig_mux]=1, then the setting USB1MODE[iddig] will control the role the controller assumes.

The loopback bit enables the loopback test. This test allows MGC1 to be connected to MGC0. It is important to set both loopback bits in both USB0/1 Mode registers. The USB0 MGC UTMI Loopback Register contains the various UTMI signals to be controlled and observed during loopback test.

The phy_test bit enables the phy_test mode. This test mode is intended to allow additional control of the UTMI inputs to the PHY. Currently, these inputs are drvvbus, dppulldown, dmpulldown, and idpullup. When phy_test is high, then the pin inputs for these signals control the inputs to the PHY instead of the Mentor controller outputs. The phy_test mode is not active with loopback mode is active.

When phy_test is active than the PHY inputs datainh is equal to datain. And txvalidh is equal to txvalid.

The USB1 mode register is shown in Figure 15-107 and described in Table 15-117.

Figure 15-107. USB1 Mode Register (USB1MODE)

31							16
Reserved							
R-0h							

15	9	8	7	6	2	1	0
Reserved		iddig	iddig_mux			phy_test	loopback
R-0h		R/W-1h	R/W-0h	R-0h		R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-117. USB1 Mode Register (USB1MODE) Field Descriptions

Bits	Field	Value	Description
31-9	Reserved	0	Always read as 0. Writes have no effect.
8	iddig		MGC input value for iddig
		0	A type
		1	B type
7	iddig_mux		Multiplexer control for IDDIG signal going to the controller.
		0	IDDIG is setting is controller by USB_ID Pin
		1	IDDIG is from bit8 (iddig) of this register
6-2	Reserved	0	Always read as 0. Writes have no effect.
1	phy_test		PHY test
		0	Normal mode
		1	PHY test mode
0	loopback		Loopback test mode
		0	Normal mode
		1	Loopback test mode

15.5.2.2.22 USB1 Mentor Core Registers

A description of the mentor core registers is available in Section 15.5.6. Two instantiations of the USB core exists, USB0 and USB1. These registers reside back to back within USB0 space. The core registers that are used by USB0 subsystem are defined within offsets 1400h-159Ch, while the core registers that are used by USB1 subsystem are defined within offsets 1C00h–1D9Ch.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.3 CPPI DMA Controller Registers

Table 15-118 lists the registers for the CPPI DMA Controller submodule.

Table 15-118. CPPI DMA Controller Registers

Address Offset	Acronym	CPPI DMA Controller Register
2000h	DMAREVID	Revision Register
2004h	TDFDQ	Teardown Free Descriptor Queue Control Register
2008h	DMAEMU	Emulation Control Register
2010h	DMAMEM1BA	CPPI Mem1 Base Address Register
2014h	DMAMEM1MASK	CPPI Mem1 Mask Address Register
2800h	TXGCR0	Tx Channel 0 Global Configuration Register
2808h	RXGCR0	Rx Channel 0 Global Configuration Register
280Ch	RXHPCRA0	Rx Channel 0 Host Packet Configuration Register A
2810h	RXHPCRB0	Rx Channel 0 Host Packet Configuration Register B
2820h	TXGCR1	Tx Channel 1 Global Configuration Register
2828h	RXGCR1	Rx Channel 1 Global Configuration Register
282Ch	RXHPCRA1	Rx Channel 1 Host Packet Configuration Register A
2830h	RXHPCRB1	Rx Channel 1 Host Packet Configuration Register B
2840h	TXGCR2	Tx Channel 2 Global Configuration Register
2848h	RXGCR2	Rx Channel 2 Global Configuration Register
284Ch	RXHPCRA2	Rx Channel 2 Host Packet Configuration Register A
2850h	RXHPCRB2	Rx Channel 2 Host Packet Configuration Register B
2860h	TXGCR3	Tx Channel 3 Global Configuration Register
2868h	RXGCR3	Rx Channel 3 Global Configuration Register
286Ch	RXHPCRA3	Rx Channel 3 Host Packet Configuration Register A
2870h	RXHPCRB3	Rx Channel 3 Host Packet Configuration Register B
2880h-2B9Fh	...	...
2BA0h	TXGCR29	Tx Channel 29 Global Configuration Register
2BA8h	RXGCR29	Rx Channel 29 Global Configuration Register
2BACh	RXHPCRA29	Rx Channel 29 Host Packet Configuration Register A
2BB0h	RXHPCRB29	Rx Channel 29 Host Packet Configuration Register B

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.3.1 CPPI DMA Revision Register (DMAREVID)

The CPPI DMA revision register (DMAREVID) contains the major and minor revisions for the module. The CPPI DMA revision register is shown in Figure 15-108 and described in Table 15-119.

Figure 15-108. CPPI DMA Revision Register (DMAREVID)

31 30	29 16	15 11	10 8	7 0
Rsvd	modID	revrtl	revmaj	revmin
R-0h	R-53h	R-2h	R-1h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-119. CPPI DMA Revision Registe (DMAREVID) Field Descriptions

Bits	Field	Description
31-30	Reserved	Always read as 0. Writes have no effect.
29-16	modID	Module ID field
15-11	revrtl	RTL revision. Will vary depending on release.
10-8	revmaj	Major revision.
7-0	revmin	Minor revision.

15.5.3.2 CPPI DMA Teardown Free Descriptor Queue Control Register (TDFDQ)

The CPPI DMA teardown free descriptor queue control register (TDFDQ) is used to inform the DMA of the location in memory or descriptor array which is to be used for signaling of a teardown complete for each Tx and Rx channel. The CPPI DMA teardown free descriptor queue control register is shown in Figure 15-109 and described in Table 15-120.

Figure 15-109. CPPI DMA Teardown Free Descriptor Queue Control Register (TDFDQ)

31 14	13 12	11 0
Reserved	td_desc_qmgr	td_desc_qnum
R-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-120. CPPI DMA Teardown Free Descriptor Queue Control Register (TDFDQ) Field Descriptions

Bits	Field	Description
31-14	Reserved	Reserved
13-12	td_desc_qmgr	This field controls which of the four Queue Managers the DMA will access in order to allocate a channel teardown descriptor from the teardown descriptor queue.
11-0	td_desc_qnum	This field controls which of the 2K queues in the indicated queue manager should be read in order to allocate channel teardown descriptors.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.3.3 CPPI DMA Emulation Control Register (DMAEMU)

The CPPI DMA emulation control register (DMAEMU) is used to control the behavior of the DMA when the emususp input is asserted. The CPPI DMA emulation control register is shown in Figure 15-110 and described in Table 15-121.

Figure 15-110. CPPI DMA Emulation Control Register (DMAEMU)

31	2	1	0
Reserved		soft	free
R-0h		R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-121. CPPI DMA Emulation Control Register (DMAEMU) Field Descriptions

Bits	Field	Value	Description
31-2	Reserved	0	Always read as 0. Writes have no effect.
1	soft		Control for emulation pause request
		0	Forces emu_pause_req low
		1	Does not force emu_pause_req low
0	free		Enable for emulation suspend

15.5.3.4 4.3.28 CPPI Mem1 Base Address Register (DMAMEM1BA)

The CPPI Mem1 base address register (DMAMEM1BA) is the base address for the CPPI mem1 accesses. This base address points to the location of the data that will be transferred from external memory to the USB DMA ram. This 16-bit field is the most significant 16-bits of the 32-bit base address.

The CPPI Mem1 base address register is shown in Figure 15-111 and described in Table 15-122.

Figure 15-111. CPPI Mem1 Base Address Register (DMAMEM1BA)

31 16	15 0
mem1_base	Reserved
R/W-0000h	R/-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-122. CPPI Mem1 Base Address Register (DMAMEM1BA) Field Descriptions

Bits	Field	Description
31-16	mem1_base	CPPI mem1 base address
15-0	Reserved	Always read as 0. Writes have no effect.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.3.5 4.3.28 CPPI Mem1 Mask Address Register (DMAMEM1MASK)

The CPPI Mem1 mask address register (DMAMEM1MASK) is the mask address for the CPPI mem1 base register. The 16-bit mem1_mask register will be ANDed with the 16-bit mem1_base register. The 16-bit resultant signal is the most significant 16-bits of the 32-bit address for external memory accesses during DMA transactions.

The CPPI Mem1 mask address register is shown in Figure 15-112 and described in Table 15-123.

Figure 15-112. CPPI Mem1 Mask Address Register (DMAMEM1MASK)

31–16	15–0
mem1_mask	Reserved
R/W-0000h	R/-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-123. CPPI Mem1 Mask Address Register (DMAMEM1MASK) Field Descriptions

Bits	Field	Description
31-16	mem1_mask	CPPI mem1 mask address
15-0	Reserved	Always read as 0. Writes have no effect.

15.5.3.6 Tx Channel N Global Configuration Register (TXGCRn)

The Tx channel N global configuration register (TXGCRn) are used to initialize the behavior of each of the Tx CPPI DMA channels. N,n ranges from 0 to 14. The Tx channel N global configuration register is shown in Figure 15-113 and described in Table 15-124.

Figure 15-113. Tx Channel N Global Configuration Register (TXGCRn)

31	30	29–16
tx_enable	tx_teardown	Reserved
R/W-0h	R/W	R

15–14	13–12	11–0
Reserved	tx_default_qmgr	tx_default_qnum
R	W	W

LEGEND: R/W = Read/Write; R = Read only; W = Write only; -*n* = value after reset

Table 15-124. Tx Channel N Global Configuration Register (TXGCRn) Field Descriptions

Bits	Field Name		Description
31	tx_enable		This field enables or disables the channel
		0	Channel is disabled
		1	Channel is enabled
			This field will be cleared after a channel teardown is complete.
30	tx_teardown		Setting this bit will request the channel to be torn down. This field will remain set after a channel teardown is complete.
29-14	Reserved		Reserved
13-12	tx_default_qmgr		This field controls the default queue manager number that will be used to queue teardown descriptors back to the host.
11-0	tx_default_qnum		This field controls the default queue number within the selected queue manager onto which teardown descriptors will be queued back to the host.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.3.7 Rx Channel N Global Configuration Register (RXGCRn)

The Rx channel global configuration registers are used to initialize the global (non descriptor type specific) behavior of each of the Rx CPPI DMA channels. If the enable bit is being set, the Rx Channel Global Configuration Register should only be written after all of the other Rx Configuration Registers have been initialized. N,n ranges from 0 to 14.

The Rx channel N global configuration register is shown in Figure 15-114 and described in Table 15-125

Figure 15-114. Rx Channel N Global Configuration Register (RXGCRn)

31	30	29 – 25	24
rx_enable	rx_teardown	Reserved	rx_error_handling
R/W-0h	R/W	R	W

23 – 16
rx_sop_offset
W

15 – 14	13 – 12	11 – 8
rx_default_desc_type	rx_default_rq_qmgr	rx_default_rq_qnum
W	W	W

7 – 0
rx_default_rq_qnum
W

LEGEND: R/W = Read/Write; R = Read only; W = Write only; -*n* = value after reset

Table 15-125. Rx Channel N Global Configuration Register (RXGCRn) Field Descriptions

Bits	Field Name	Value	Description
31	rx_enable		This field enables or disables the channel
		0	Channel is disabled
		1	Channel is enabled
			This field will be cleared after a channel teardown is complete.
30	rx_teardown		This field indicates whether or not an Rx teardown operation is complete. This field should be cleared when a channel is initialized. This field will be set after a channel teardown is complete.
29-25	Reserved		Reserved
24	rx_error_handling		This bit controls the error handling mode for the channel and is only used when channel errors (i.e. descriptor or buffer starvation occurs):
		0	Starvation errors result in dropping packet and reclaiming any used descriptor or buffer resources back to the original queues/pools they were allocated to
		1	Starvation errors result in subsequent re-try of the descriptor allocation operation. In this mode, the DMA will return to the IDLE state without saving it's internal operational state back to the internal state RAM and without issuing an advance operation on the FIFO interface. This results in the DMA re-initiating the FIFO block transfer at a later time with the intention that additional free buffers and/or descriptors will have been added.
			Regardless of the value of this bit, the DMA will assert the cdma_rx_sof_overrun (for SOP) or cdma_rx_mof_overrun (for non-SOP) when
23:16	rx_sop_offset		This field specifies the number of bytes that are to be skipped in the SOP buffer before beginning to write the payload. This value must be less than the minimum size of a buffer in the system. Valid values are 0 – 255 bytes.
15-14	rx_default_desc_type		This field indicates the default descriptor type to use:
		0	Reserved
		1	Host
		2	Reserved
		3	Reserved
			The actual descriptor type that will be used for reception can be overridden by information provided in the CPPI FIFO data block.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-125. Rx Channel N Global Configuration Register (RXGCRn) Field Descriptions (continued)

Bits	Field Name	Value	Description
13-12	rx_default_rq_qmgr		This field indicates the default receive queue manager that this channel should use. The actual receive queue manager index can be overridden by information provided in the CPPI FIFO data block.
11-0	rx_default_rq_qnum		This field indicates the default receive queue that this channel should use. The actual receive queue that will be used for reception can be overridden by information provided in the CPPI FIFO data block.

15.5.3.8 Rx Channel N Host Packet Configuration Register A (RXHPCRAn)

The Rx channel host packet configuration A registers are used to initialize the behavior of each of the Rx CPPI DMA channels for reception of host type packets. N,n ranges from 0 to 14.

The Rx channel host packet configuration A registers are shown in Figure 15-115 and described in Table 15-126

Figure 15-115. Rx Channel N Host Packet Configuration Register A (RXHPCRAn)

31 30	29 28	27 16
Reserved	rx_host_fdq1_qmgr	rx_host_fdq1_qnum
R-0	W	W

15 14	13 12	11 0
Reserved	rx_host_fdq0_qmgr	rx_host_fdq0_qnum
R-0	W	W

LEGEND: R/W = Read/Write; R = Read only; W = Write only; -*n* = value after reset

Table 15-126. Rx Channel N Host Packet Configuration Register A (RXHPCRAn) Field Descriptions

Bits	Field Name	Description
31-30	Reserved	Reserved
29-28	rx_host_fdq1_qmgr	This field specifies which Buffer Manager should be used for the second Rx buffer in a host type packet.
27-16	rx_host_fdq1_qnum	This field specifies which Free Descriptor / Buffer Pool should be used for the second Rx buffer in a host type packet.
15-14	Reserved	Reserved
13-12	rx_host_fdq0_qmgr	This field specifies which Buffer Manager should be used for the first Rx buffer in a host type packet.
11-0	rx_host_fdq0_qnum	This field specifies which Free Descriptor / Buffer Pool should be used for the frst Rx buffer in a host type packet.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.3.9 Rx Channel N Host Packet Configuration Register B (RXHPCRBn)

The Rx channel host packet configuration B registers are used to initialize the behavior of each of the Rx CPPI DMA channels for reception of host type packets. N,n ranges from 0 to 14.

The Rx channel host packet configuration B registers are shown in Figure 15-116 and described in Table 15-127

Figure 15-116. Rx Channel N Host Packet Configuration Register B (RXHPCRBn)

31 30	29 28	27 16
Reserved	rx_host_fdq3_qmgr	rx_host_fdq3_qnum
R-0	W	W

15 14	13 12	11 0
Reserved	rx_host_fdq2_qmgr	rx_host_fdq2_qnum
R-0	W	W

LEGEND: R/W = Read/Write; R = Read only; W = Write only; -*n* = value after reset

Table 15-127. Rx Channel N Host Packet Configuration Register B (RXHPCRBn) Field Descriptions

Bits	Field Name	Description
31-30	Reserved	Reserved
29-28	rx_host_fdq3_qmgr	This field specifies which Manager should be used for the fourth or later Rx buffers in a host type packet.
27-16	rx_host_fdq3_qnum	This field specifies which Free Descriptor Queue should be used for the fourth or later Rx buffers in a host type packet.
15-14	Reserved	Reserved
13-12	rx_host_fdq2_qmgr	This field specifies which Buffer Manager should be used for the third Rx buffer in a host type packet.
11-0	rx_host_fdq2_qnum	This field specifies which Free Descriptor / Buffer Pool should be used for the third Rx buffer in a host type packet.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.4 CPPI DMA Scheduler Registers

Table 15-128 lists the registers for the CPPI DMA Scheduler submodule.

Table 15-128. CPPI DMA Scheduler Registers

Address Offset	Acronym	CPPI DMA Scheduler Register
3000h	DMA_SCHED_CTRL	CPPI DMA Scheduler Control Register
3800h-38FCh	WORD0-WORD63	CPPI DMA Scheduler Table Word 0-Word 63

15.5.4.1 CPPI DMA Scheduler Control Register (DMA_SCHED_CTRL)

The CPPI DMA scheduler control revision register contains the major and minor revisions for the module. The revision register is shown in Figure 15-117 and described in Table 15-129.

Figure 15-117. CPPI DMA Scheduler Control Register (DMA_SCHED_CTRL)

31	30 … 8	7 … 0
enable	Reserved	last_entry
R/W-0h	R-07-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-129. CPPI DMA Scheduler Control Register (DMA_SCHED_CTRL) Field Descriptions

Bits	Field Name	Value	Description
31	enable		This is the enable bit for the scheduler and is encoded as follows:
		0	Scheduler is disabled and will no longer fetch entries from the scheduler table or pass credits to the DMA controller.
		1	Scheduler is enabled.
			This bit should only be set after the table has been initialized.
30-8	Reserved		Reserved
7-0	last_entry		This field indicates the last valid entry in the scheduler table. There are 64 words in the table and there are 4 entries in each word. The table can be programmed with any integer number of entries from 1 to 256. The corresponding encoding for this field is as follows:
		0	1 entry
		1	2 entries
		...	…
		254	255 entries
		255	256 entries

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.4.2 CPPI DMA Scheduler Table Word N (WORDn)

The CPPI DMA scheduler table word N (WORDn) registers are used to initialize the behavior of each of the Tx DMA channels. N ranges from 0 to 63.

The CPPI DMA scheduler table word N (WORDn) registers are shown in Figure 15-118 and described in Table 15-130.

Figure 15-118. CPPI DMA Scheduler Table Word N (WORDn)

31	30	29	28	24
entry3_rxtx	Reserved		entry3_channel	
W	R		R	W

23	22	21	20	16
entry2_rxtx	Reserved		entry2_channel	
W	R		W	

15	14	13	12	8
entry1_rxtx	Reserved		entry1_channel	
W	R		W	

7	6	5	4	0
entry0_rxtx	Reserved		entry0_channel	
W	R		W	

LEGEND: R/W = Read/Write; R = Read only; W = Write only; -*n* = value after reset

Table 15-130. CPPI DMA Scheduler Table Word N (WORDn) Field Descriptions

Bits	Field Name	Value	Description
31	entry3_rxtx		This bit indicates if this entry is for a Tx or an Rx channel and is encoded as follows:
		0	Tx Channel
		1	Rx Channel
30-29	Reserved		Reserved
28-24	entry3_channel		This field indicates the channel number that is to be given an opportunity to transfer data. If this is a Tx entry, the DMA will be presented with a scheduling 'credit' for that exact Tx channel. If this is an Rx entry, the DMA will be presented with a scheduling credit for the Rx FIFO that is associated with this channel. For Rx FIFOs which carry traffic for more than 1 Rx DMA channel, the exact channel number that is given in the Rx credit will actually be the channel number which is currently on the head element of that Rx FIFO, which is not necessarily the channel number given in the scheduler table entry.
23	entry2_rxtx		This bit indicates if this entry is for a Tx or an Rx channel and is encoded as follows:
		0	Tx Channel
		1	Rx Channel
22-21	Reserved		Reserved
20-16	entry2_channel		This field indicates the channel number that is to be given an opportunity to transfer data. If this is a Tx entry, the DMA will be presented with a scheduling 'credit' for that exact Tx channel. If this is an Rx entry, the DMA will be presented with a scheduling credit for the Rx FIFO that is associated with this channel. For Rx FIFOs which carry traffic for more than 1 Rx DMA channel, the exact channel number that is given in the Rx credit will actually be the channel number which is currently on the head element of that Rx FIFO, which is not necessarily the channel number given in the scheduler table entry.
15	entry1_rxtx		This bit indicates if this entry is for a Tx or an Rx channel and is encoded as follows:
		0	Tx Channel
		1	Rx Channel
14:-13	Reserved		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-130. CPPI DMA Scheduler Table Word N (WORDn) Field Descriptions (continued)

Bits	Field Name	Value	Description
12-8	entry1_channel		This field indicates the channel number that is to be given an opportunity to transfer data. If this is a Tx entry, the DMA will be presented with a scheduling 'credit' for that exact Tx channel. If this is an Rx entry, the DMA will be presented with a scheduling credit for the Rx FIFO that is associated with this channel. For Rx FIFOs which carry traffic for more than 1 Rx DMA channel, the exact channel number that is given in the Rx credit will actually be the channel number which is currently on the head element of that Rx FIFO, which is not necessarily the channel number given in the scheduler table entry.
7	entry0_rxtx		This bit indicates if this entry is for a Tx or an Rx channel and is encoded as follows:
		0	Tx Channel
		1	Rx Channel
6-5	Reserved		Reserved
4-0	entry0_channel		This field indicates the channel number that is to be given an opportunity to transfer data. If this is a Tx entry, the DMA will be presented with a scheduling 'credit' for that exact Tx channel. If this is an Rx entry, the DMA will be presented with a scheduling credit for the Rx FIFO that is associated with this channel. For Rx FIFOs which carry traffic for more than 1 Rx DMA channel, the exact channel number that is given in the Rx credit will actually be the channel number which is currently on the head element of that Rx FIFO, which is not necessarily the channel number given in the scheduler table entry.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.5 CPPI DMA Queue Manager Registers

Table 15-131 lists the registers for the CPPI DMA Queue Manager submodule.

Table 15-131. CPPI DMA Queue Manager Registers

Address Offset	Queue Manager Register
4000h	Qmgr Revision Register
4004h	Qmgr Reset Register
4008h	Qmgr Queue Diversion Register
4020h	Qmgr Free Descriptor/Buffer Starvation Count Qmgr Register 0
4024h	Qmgr Free Descriptor/Buffer Starvation Count Register 1
4028h	Qmgr Free Descriptor/Buffer Starvation Count Register 2
402Ch	Qmgr Free Descriptor/Buffer Starvation Count Register 3
4030h	Qmgr Free Descriptor/Buffer Starvation Count Qmgr Register 4
4034h	Qmgr Free Descriptor/Buffer Starvation Count Register 5
4038h	Qmgr Free Descriptor/Buffer Starvation Count Register 6
403Ch	Qmgr Free Descriptor/Buffer Starvation Count Register 7
4080h	Qmgr Linking RAM Region 0 Base Address Register
4084h	Qmgr Linking RAM Region 0 Size Register
4088h	Qmgr Linking RAM Region 1 Base Address Register
4090h	Qmgr Queue Pending Register 0
4094h	Qmgr Queue Pending Register 1
4098h	Qmgr Queue Pending Register 2
409Ch	Qmgr Queue Pending Register 3
40A0h	Qmgr Queue Pending Register 4
5000h + 16 x R	Memory Region R Base Address Register
5000h + 16 x R + 4	Memory Region R Control Register
6000h + 16 x N	Queue Manager Queue N Register A
6004h + 16 x N	Queue Manager Queue N Register B
6008h + 16 x N	Queue Manager Queue N Register C
600Ch + 16 x N	Queue N Register D
7000h + 16 x N	Queue N Status Register A
7004h + 16 x N	Queue N Status Register B
7008h + 16 x N	Queue N Status Register C

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.5.1 Queue Manager Revision Register (QMGRREVID)

The queue manager revision register contains the major and minor revisions for the module. It does not support byte accesses.

The revision register is shown in Figure 15-119 and described in Table 15-132.

Figure 15-119. Queue Manager Revision Register (QMGRREVID)

31–30	29–28	27–16
Scheme	Reserved	function
R/1h	R-0h	R-e53h

15–11	10–8	7–6	5–0
revrtl	revmaj	revcustom	revmin
R/1h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-132. Queue Manager Revision Register (QMGRREVID) Field Descriptions

Bits	Field Name	Description
31-30	scheme	Scheme that this register is compliant with
29-28	Reserved	Reserved
27-16	function	Function
15-11	revrtl	RTL revision
10-8	revmaj	Major revision
7-6	revcustom	Custom revision
5-0	revmin	Minor revision

15.5.5.2 Queue Manager Reset Register (QMGRRST)

The queue manager reset register is used to transfer the contents of one queue onto another queue. It does not support byte accesses.

The queue manager reset register is shown in Figure 15-120 and described in Table 15-132.

Figure 15-120. Queue Manager Reset Register (QMGRRST)

31	30	29–16
head_tail	Rsvd	dest_qnum
W-0h	R-0h	W-0h

15–14	13–0
Reserved	source_qnum
R-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-133. Queue Manager Reset Register (QMGRRST) Field Descriptions

Bits	Field Name	Description
31	head_tail	Indicates whether queue contents should be merged on to head or tail of destination queue. Clear this field for head and set for tail.
30	Reserved	Reserved
29-16	dest_qnum	Destination Queue Number
15-14	Reserved	Reserved
13-0	source_qnum	Source Queue Number

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.5.3 Queue Manager Queue Diversion Register (DIVERSION)

The queue manager queue diversion register is used to transfer the contents of one queue onto another queue. It does not support byte accesses.

The queue manager queue diversion register is shown in Figure 15-121 and described in Table 15-134.

Figure 15-121. Queue Manager Queue Diversion Register (DIVERSION)

31	30	29 … 16
head_tail	Rsvd	dest_qnum
W-0h	R-0h	W-0h

15 … 14	13 … 0
Reserved	source_qnum
R-0h	W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-134. Queue Manager Queue Diversion Register (DIVERSION) Field Descriptions

Bits	Field Name	Description
31	head_tail	Indicates whether queue contents should be merged on to head or tail of destination queue. Clear this field for head and set for tail.
30	Reserved	Reserved
29-16	dest_qnum	Destination Queue Number
15-14	Reserved	Reserved
13-0	source_qnum	Source Queue Number

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.5.4 Queue Manager Free Descriptor/Buffer Starvation Count Register 0 (FDBSC0)

The free descriptor/buffer queue starvation count register 0 provides statistics about how many starvation events are occurring on the Rx free descriptor/buffer queues 0-3. It does not support byte accesses.

The free descriptor/buffer queue starvation count register 0 is shown in Figure 15-122 and described in Table 15-135.

Figure 15-122. Queue Manager Free Descriptor/Buffer Starvation Count Register 0 (FDBSC0)

31–24	23–16	15–8	7–0
fdbq3_starve_cnt	fdbq2_starve_cnt	fdbq1_starve_cnt	fdbq0_starve_cnt
RC-0h	RC-0h	RC-0h	RC-0h

LEGEND: R/W = Read/Write; R = Read only; RC = Read Clear; -*n* = value after reset

Table 15-135. Queue Manager Free Descriptor/Buffer Starvation Count Register 0 (FDBSC0) Field Descriptions

Bits	Field Name	Description
31-24	fdbq3_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 3 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.
23-16	fdbq2_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 2 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.
15-8	fdbq1_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 1 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.
7-0	fdbq0_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 0 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.

15.5.5.5 Queue Manager Free Descriptor/Buffer Starvation Count Register 1 (FDBSC1)

The free descriptor/buffer queue starvation count register 1 provides statistics about how many starvation events are occurring on the Rx free descriptor/buffer queues 4-7. It does not support byte accesses.

The free descriptor/buffer queue starvation count register 1 is shown in Figure 15-123 and described in Table 15-136.

Figure 15-123. Queue Manager Free Descriptor/Buffer Starvation Count Register 1 (FDBSC1)

31–24	23–16	15–8	7–0
fdbq7_starve_cnt	fdbq6_starve_cnt	fdbq5_starve_cnt	fdbq4_starve_cnt
RC-0h	RC-0h	RC-0h	RC-0h

LEGEND: R/W = Read/Write; R = Read only; RC = Read Clear; -*n* = value after reset

Table 15-136. Queue Manager Free Descriptor/Buffer Starvation Count Register 1 (FDBSC1) Field Descriptions

Bits	Field Name	Description
31-24	fdbq7_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 7 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.
23-16	fdbq6_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 6 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.
15-8	fdbq5_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 5 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.
7-0	fdbq4_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 4 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.5.6 Queue Manager Free Descriptor/Buffer Starvation Count Register 2 (FDBSC2)

The free descriptor/buffer queue starvation count register 2provides statistics about how many starvation events are occurring on the Rx free descriptor/buffer queues 8-11. It does not support byte accesses.

The free descriptor/buffer queue starvation count register 2 is shown in Figure 15-124 and described in Table 15-137.

Figure 15-124. Queue Manager Free Descriptor/Buffer Starvation Count Register 2 (FDBSC2)

31–24	23–16	15–8	7–0
fdbq11_starve_cnt	fdbq10_starve_cnt	fdbq9_starve_cnt	fdbq8_starve_cnt
RC-0h	RC-0h	RC-0h	RC-0h

LEGEND: R/W = Read/Write; R = Read only; RC = Read Clear; -*n* = value after reset

Table 15-137. Queue Manager Free Descriptor/Buffer Starvation Count Register 2 (FDBSC2) Field Descriptions

Bits	Field Name	Description
31-24	fdbq11_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 11 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.
23-16	fdbq10_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 10 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.
15-8	fdbq9_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 9 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.
7-0	fdbq8_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 8 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.

15.5.5.7 Queue Manager Free Descriptor/Buffer Starvation Count Register 3 (FDBSC3)

The free descriptor/buffer queue starvation count register 3 provides statistics about how many starvation events are occurring on the Rx free descriptor/buffer queues 12-15. It does not support byte accesses.

The free descriptor/buffer queue starvation count register 3 is shown in Figure 15-125 and described in Table 15-138.

Figure 15-125. Queue Manager Free Descriptor/Buffer Starvation Count Register 3 (FDBSC3)

31–24	23–16	15–8	7–0
fdbq15_starve_cnt	fdbq14_starve_cnt	fdbq13_starve_cnt	fdbq12_starve_cnt
RC-0h	RC-0h	RC-0h	RC-0h

LEGEND: R/W = Read/Write; R = Read only; RC = Read Clear; -*n* = value after reset

Table 15-138. Queue Manager Free Descriptor/Buffer Starvation Count Register 3 (FDBSC3) Field Descriptions

Bits	Field Name	Description
31-24	fdbq15_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 15 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.
23-16	fdbq14_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 14 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.
15-8	fdbq13_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 13 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.
7-0	fdbq12_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 12 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.5.8 Queue Manager Free Descriptor/Buffer Starvation Count Register 4 (FDBSC4)

The free descriptor/buffer queue starvation count register 4 provides statistics about how many starvation events are occurring on the Rx free descriptor/buffer queues 16-19. It does not support byte accesses.

The free descriptor/buffer queue starvation count register 4 is shown in Figure 15-126 and described in Table 15-139.

Figure 15-126. Queue Manager Free Descriptor/Buffer Starvation Count Register 4 (FDBSC4)

31–24	23–16	15–8	7–0
fdbq19_starve_cnt	fdbq18_starve_cnt	fdbq17_starve_cnt	fdbq16_starve_cnt
RC-0h	RC-0h	RC-0h	RC-0h

LEGEND: R/W = Read/Write; R = Read only; RC = Read Clear; -*n* = value after reset

Table 15-139. Queue Manager Free Descriptor/Buffer Starvation Count Register 4 (FDBSC4) Field Descriptions

Bits	Field Name	Description
31-24	fdbq19_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 19 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.
23-16	fdbq18_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 18 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.
15-8	fdbq17_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 17 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.
7-0	fdbq16_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 16 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.

15.5.5.9 Queue Manager Free Descriptor/Buffer Starvation Count Register 5 (FDBSC5)

The free descriptor/buffer queue starvation count register 5 provides statistics about how many starvation events are occurring on the Rx free descriptor/buffer queues 20-23. It does not support byte accesses.

The free descriptor/buffer queue starvation count register 5 is shown in Figure 15-127 and described in Table 15-140.

Figure 15-127. Queue Manager Free Descriptor/Buffer Starvation Count Register 5 (FDBSC5)

31–24	23–16	15–8	7–0
fdbq23_starve_cnt	fdbq22_starve_cnt	fdbq21_starve_cnt	fdbq20_starve_cnt
RC-0h	RC-0h	RC-0h	RC-0h

LEGEND: R/W = Read/Write; R = Read only; RC = Read Clear; -*n* = value after reset

Table 15-140. Queue Manager Free Descriptor/Buffer Starvation Count Register 5 (FDBSC5) Field Descriptions

Bits	Field Name	Description
31-24	fdbq23_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 23 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.
23-16	fdbq22_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 22 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.
15-8	fdbq21_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 21 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.
7-0	fdbq20_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 20 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.5.10 Queue Manager Free Descriptor/Buffer Starvation Count Register 6 (FDBSC6)

The free descriptor/buffer queue starvation count register 6 provides statistics about how many starvation events are occurring on the Rx free descriptor/buffer queues 24-27. It does not support byte accesses.

The free descriptor/buffer queue starvation count register 6 is shown in Figure 15-128 and described in Table 15-141.

Figure 15-128. Queue Manager Free Descriptor/Buffer Starvation Count Register 6 (FDBSC6)

31 – 24	23 – 16	15 – 8	7 – 0
fdbq27_starve_cnt	fdbq26_starve_cnt	fdbq25_starve_cnt	fdbq24_starve_cnt
RC-0h	RC-0h	RC-0h	RC-0h

LEGEND: R/W = Read/Write; R = Read only; RC = Read Clear; -*n* = value after reset

Table 15-141. Queue Manager Free Descriptor/Buffer Starvation Count Register 6 (FDBSC6) Field Descriptions

Bits	Field Name	Description
31-24	fdbq27_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 27 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.
23-16	fdbq26_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 26 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.
15-8	fdbq25_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 25 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.
7-0	fdbq24_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 24 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.

15.5.5.11 Queue Manager Free Descriptor/Buffer Starvation Count Register 7 (FDBSC7)

The free descriptor/buffer queue starvation count register 7 provides statistics about how many starvation events are occurring on the Rx free descriptor/buffer queues 28-31. It does not support byte accesses.

The free descriptor/buffer queue starvation count register 7 is shown in Figure 15-129 and described in Table 15-142.

Figure 15-129. Queue Manager Free Descriptor/Buffer Starvation Count Register 7 (FDBSC7)

31 – 24	23 – 16	15 – 8	7 – 0
fdbq31_starve_cnt	fdbq30_starve_cnt	fdbq29_starve_cnt	fdbq28_starve_cnt
RC-0h	RC-0h	RC-0h	RC-0h

LEGEND: R/W = Read/Write; R = Read only; RC = Read Clear; -*n* = value after reset

Table 15-142. Queue Manager Free Descriptor/Buffer Starvation Count Register 7 (FDBSC7) Field Descriptions

Bits	Field Name	Description
31-24	fdbq31_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 31 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.
23-16	fdbq30_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 30 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.
15-8	fdbq29_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 29 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.
7:0	fdbq28_starve_cnt	This field increments each time the Free Descriptor/Buffer Queue 28 is read while it is empty via the CPPI DMA. This field is cleared when read via the CPU.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.5.12 Queue Manager Linking RAM Region 0 Base Address Register (LRAM0BASE)

The Linking RAM region 0 base address register is used to set the base address for the first portion of the Linking RAM. This address must be 32-bit aligned. It is used by the Queue Manager to calculate the 32-bit linking address for a given descriptor index. It does not support byte accesses.

The Linking RAM region 0 base address register is shown in Figure 15-130 and described in Table 15-143.

Figure 15-130. Queue Manager Linking RAM Region 0 Base Address Register (LRAM0BASE)

31	0
region0_base	
R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; RC = Read Clear; -*n* = value after reset

Table 15-143. Queue Manager Linking RAM Region 0 Base Address Register (LRAM0BASE) Field Descriptions

Bits	Field Name	Description
31-0	region0_base	This field stores the base address for the first region of the linking RAM. This may be anywhere in 32-bit address space but would be typically located in on-chip memory.

15.5.5.13 Queue Manager Linking RAM Region 0 Size Register (LRAM0SIZE)

The Linking RAM region 0 size register is used to set the size of the array of linking pointers that are located in Region 0 of Linking RAM. The size specified the number of descriptors for which linking information is stored in this region. It does not support byte accesses.

The Linking RAM region 0 size register is shown in Figure 15-131 and described in Table 15-144.

Figure 15-131. Queue Manager Linking RAM Region 0 Size Register (LRAM0SIZE)

31 14	13 0
Reserved	region0_size
R-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-144. Queue Manager Linking RAM Region 0 Size Register (LRAM0SIZE) Field Descriptions

Bits	Field Name	Description
31-14	Reserved	Reserved
13-0	region0_size	This field indicates the number of entries that are contained in the linking RAM region 0. A descriptor with index less than region0_size value has its linking location in region 0. A descriptor with index greater than region0_size has its linking location in region 1. The queue manager will add the index (left shifted by 2 bits) to the appropriate regionX_base_addr to get the absolute 32-bit address to the linking location for a descriptor.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.5.14 Queue Manager Linking RAM Region 1 Base Address Register (LRAM1BASE)

The Linking RAM region 1 base address register is used to set the base address for the second portion of the Linking RAM. This base address is used by the Queue Manager to calculate the 32-bit linking address from the descriptor index. All descriptors with index higher than that given in Linking RAM 0 Size register have linking information stored in linking RAM region 1. It does not support byte accesses.

The Linking RAM region 1 base address register is shown in Figure 15-132 and described in Table 15-145.

Figure 15-132. Queue Manager Linking RAM Region 1 Base Address Register (LRAM1BASE)

31	0
region1_base	
R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-145. Queue Manager Linking RAM Region 1 Base Address Register (LRAM1BASE) Field Descriptions

Bits	Field Name	Description
31-0	region1_base	This field stores the base address for the second region of the linking RAM. This may be anywhere in 32-bit address space but would be typically located in off-chip memory.

15.5.5.15 Queue Manager Queue Pending Register 0 (PEND0)

The queue pending register 0 can be read to find the pending status for queues 31 to 0. It does not support byte accesses.

The queue pending register 0 is shown in Figure 15-133 and described in Table 15-146.

Figure 15-133. Queue Manager Queue Pending Register 0 (PEND0)

31	0
qpend0	
R-0h	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-146. Queue Manager Queue Pending Register 0 (PEND0) Field Descriptions

Bits	Field Name	Description
31-0	qpend0	This field indicates the queue pending status for queues[31:0].

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.5.16 Queue Manager Queue Pending Register 1 (PEND1)

The queue pending register 1 can be read to find the pending status for queues 63 to 32. It does not support byte accesses.

The queue pending register 1 is shown in Figure 15-134 and described in Table 15-147.

Figure 15-134. Queue Manager Queue Pending Register 1 (PEND1)

31	0
qpend1	
R-0h	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-147. Queue Manager Queue Pending Register 1 (PEND1) Field Descriptions

Bits	Field Name	Description
31-0	qpend1	This field indicates the queue pending status for queues[63:32]

15.5.5.17 Queue Manager Queue Pending Register 2 (PEND2)

The queue pending register 2 can be read to find the pending status for queues 95 to 64. It does not support byte accesses.

The queue pending register 2 is shown in Figure 15-135 and described in Table 15-148.

Figure 15-135. Queue Manager Queue Pending Register 2 (PEND2)

31	0
qpend2	
R-0h	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-148. Queue Manager Queue Pending Register 2 (PEND1) Field Descriptions

Bits	Field Name	Description
31-0	qpend2	This field indicates the queue pending status for queues[95:64]

15.5.5.18 Queue Manager Queue Pending Register 3 (PEND3)

The queue pending register 3 can be read to find the pending status for queues 127 to 96. It does not support byte accesses.

The queue pending register 3 is shown in Figure 15-136 and described in Table 15-149.

Figure 15-136. Queue Manager Queue Pending Register 3 (PEND3)

31	0
qpend3	
R-0h	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-149. Queue Manager Queue Pending Register 3 (PEND3) Field Descriptions

Bits	Field Name	Description
31-0	qpend3	This field indicates the queue pending status for queues[127:96]

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.5.19 Queue Manager Queue Pending Register 4 (PEND4)

The queue pending register 4 can be read to find the pending status for queues 159 to 128. It does not support byte accesses.

The queue pending register 4 is shown in Figure 15-137 and described in Table 15-150.

Figure 15-137. Queue Manager Queue Pending Register 4 (PEND4)

31	0
qpend4	
R-0h	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-150. Queue Manager Queue Pending Register 4 (PEND4) Field Descriptions

Bits	Field Name	Description
31-0	qpend4	This field indicates the queue pending status for queues[159:128]

15.5.5.20 Queue Manager Memory Region R Base Address Register (QMEMRBASEr)

The memory region R base address register is written by the host to set the base address of memory region R. This memory region will store a number of descriptors of a particular size as determined by the Memory Region R Control Register. It does not support byte accesses. R ranges from 0 to 15.

The memory region R base address register is shown in Figure 15-138 and described in Table 15-151.

Figure 15-138. Queue Manager Memory Region R Base Address Register (QMEMRBASEr)

31	0
reg	
R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-151. Queue Manager Memory Region R Base Address Register (QMEMRBASEr) Field Descriptions

Bits	Field Name	Description
31-0	reg	This field contains the base address of the memory region R.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.5.21 Queue Manager Memory Region R Control Register (QMEMRCTRLr)

The memory region R control register is written by the host to configure various parameters of this memory region. It does not support byte accesses. R ranges from 0 to 15.

The memory region R control register is shown in Figure 15-139 and described in Table 15-152.

Figure 15-139. Queue Manager Memory Region R Control Register (QMEMRCTRLr)

31 30	29 16	15 12	11 8	7 3	2 0
Rsvd	start_index	Reserved	desc_size	Reserved	reg_size
R-0h	R/W-0h	R-0h	R/W-0h	R-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-152. Queue Manager Memory Region R Control Register (QMEMRCTRLr) Field Descriptions

Bits	Field Name	Description	Type	Reset Value
31-30	Reserved	Reserved	R	0h
29-16	start_index	This field indicates where in linking RAM does the descriptor linking information corresponding to memory region R starts.	R/W	0h
15-12	Reserved	Reserved	R	0h
11:08	desc_size	This field indicates the size of each descriptor in this memory region. It is an encoded value that specifies descriptor size as 2^(5+desc_size) number of bytes. The settings of desc_size from 9-15 are reserved.	R/W	0h
7-3	Reserved	Reserved	R	0h
2-0	reg_size	This field indicates the size of the memory region (in terms of number of descriptors). It is an encoded value that specifies region size as 2^(5+reg_size) number of descriptors.	R/W	0h

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.5.22 Queue Manager Queue N Register A (CTRLAn)

The queue N register A is an optional register that is only implemented for a queue if the queue supports entry/byte count feature. The entry count feature provides a count of the number of entries that are currently valid in the queue. It does not support byte accesses. N ranges from 0 to 155.

The queue N register A is shown in Figure 15-140 and described in Table 15-153.

Figure 15-140. Queue Manager Queue N Register A (CTRLAn)

31–14	13–0
Reserved	queue_entry_count
R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-153. Queue Manager Queue N Register A (CTRLAn) Field Descriptions

Bits	Field Name	Description
31-14	Reserved	Reserved
13-0	queue_entry_count	This field indicates how many packets are currently queued on the queue. This count is incremented by 1 whenever a packet is added to the queue. This count is decremented by 1 whenever a packet is popped from the queue.

15.5.5.23 Queue Manager Queue N Register B (CTRLBn)

The Queue N Register B is an optional register that is only implemented for a queue if the queue supports a total byte count feature. The total byte count feature provides a count of the total number of bytes in all of the packets that are currently valid in the queue. This register must be read prior to reading Queue N register D during packet pop operation if the total size information is desired. It does not support byte accesses. N ranges from 0 to 155.

The queue N register B is shown in Figure 15-141 and described in Table 15-154.

Figure 15-141. Queue Manager Queue N Register B (CTRLBn)

31–28	27–0
Reserved	queue_byte_count
R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-154. Queue Manager Queue N Register B (CTRLBn) Field Descriptions

Bits	Field Name	Description
31-28	Reserved	Reserved
27-0	queue_byte_count	This field indicates how many bytes total are contained in all of the packets which are currently queued on this queue.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.5.24 Queue Manager Queue N Register C (CTRLCn)

The Queue N Register C is used to provide additional information about the packet that is being pushed or popped from the queue. This register provides an option for the packet to be pushed onto either the tail of the queue (default) or the head of the queue (override). This register must be written prior to writing the Queue N register D during packet write operations. This register must be read prior to reading Queue N register D during pop operations if the packet size information is desired. It does not support byte accesses. N ranges from 0 to 155.

The queue N register C is shown in Figure 15-142 and described in Table 15-155.

Figure 15-142. Queue Manager Queue N Register C (CTRLCn)

31 – 14	13 – 0
Reserved	packet_size
R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-155. Queue Manager Queue N Register C (CTRLCn) Field Descriptions

Bits	Field Name	Description
31-14	Reserved	Reserved
13-0	packet_size	This field indicates packet size of the head element of a queue. This field indicates packet size for packet pop operation.

15.5.5.25 Queue Manager Queue N Register D (CTRLDn)

The queue N register D is written to add a packet to the queue and read to pop a packets off a queue. The packet is only pushed or popped to/from the queue when the queue register D is written. It does not support byte accesses. N ranges from 0 to 155.

The queue N register D is shown in Figure 15-143 and described in Table 15-156.

Figure 15-143. Queue Manager Queue N Register D (CTRLDn)

31 – 5	4 – 0
desc_ptr	desc_size
R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-156. Queue Manager Queue N Register D (CTRLDn) Field Descriptions

Bits	Field Name	Description
31-5	desc_ptr	Descriptor pointer. It will be read as zero if the queue is empty. It will indicate a 32-bit aligned address that points to a descriptor when the queue is not empty.
4-0	desc_size	Descriptor Size. It is encoded in 4-byte increments with values 0 to 31 representing 24 and so on to 148 bytes. This field will return a 0x0 when an empty queue is read.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.5.26 Queue Manager Queue N Status Register A (QSTATAn)

The Queue N Status Register A is an optional register that is only implemented for a queue if the queue supports entry/byte count feature. The entry count feature provides a count of the number of entries that are currently valid in the queue. It does not support byte accesses. N ranges from 0 to 155.

The queue N status register A is shown in Figure 15-144 and described in Table 15-157.

Figure 15-144. Queue Manager Queue N Register A (QSTATAn)

31–14	13–0
Reserved	queue_entry_count
R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-157. Queue Manager Queue N Status Register A (QSTATAn) Field Descriptions

Bits	Field Name	Description
31-14	Reserved	Reserved
13-0	queue_entry_count	This field indicates how many packets are currently queued on the queue.

15.5.5.27 Queue Manager Queue N Status Register B (QSTATBn)

The Queue N Status Register B is an optional register that is only implemented for a queue if the queue supports a total byte count feature. The total byte count feature provides a count of the total number of bytes in all of the packets that are currently valid in the queue. It does not support byte accesses. N ranges from 0 to 155.

The queue N status register B is shown in Figure 15-145 and described in Table 15-158.

Figure 15-145. Queue Manager Queue N Status Register B (QSTATBn)

31–28	27–0
Reserved	queue_byte_count
R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-158. Queue Manager Queue N Status Register B (QSTATBn) Field Descriptions

Bits	Field Name	Description
31-28	Reserved	Reserved
27-0	queue_byte_count	This field indicates how many bytes total are contained in all of the packets which are currently queued on this queue.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.5.28 Queue Manager Queue N Status Register C (QSTATCn)

The Queue N Status Register C specifies the packet size for the head element of a queue. It does not support byte accesses. N ranges from 0 to 155.

The queue N status register C is shown in Figure 15-146 and described in Table 15-159.

Figure 15-146. Queue Manager Queue N Status Register C (QSTATCn)

31 – 14	13 – 0
Reserved	packet_size
R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-159. Queue Manager Queue N Status Register C (QSTATCn) Field Descriptions

Bits	Field Name	Description
31-14	Reserved	Reserved
13-0	packet_size	This field indicates packet size of the head element of a queue.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.6 USB Mentor Core Registers

The Mentor Control register map is divided into the following sections:

- **Common USB registers (0h–Fh)** – These registers provide control and status for the complete core.
- **Indexed Endpoint Control/Status registers (10h–1Fh)** – These registers provide control and status for the currently selected endpoint. The registers mapped into this section depend on whether the core is in Peripheral mode or in Host mode and on the value of the Index register.
- **FIFOs (20h–5Fh)** – This address range provides access to the endpoint FIFOs.
- Additional Control and Configuration registers (60h–7Fh) – These registers provide additional device status and control.
- **Target Endpoint Control Registers (80h–FFh)** – These registers provide target function and hub address details for each of the endpoints.
- **Non-Indexed Endpoint Control/Status registers (100h and above)** – The registers available at 10h–1Fh, accessible independently of the setting of the Index register. 100h–10Fh, EP0 registers; 110h–11Fh, EP1 registers; 120h–12Fh, EP2; and so on.

Two instances of the Mentor Core registers exist within USB0 space. Since both USB modules operate independent of each other, each USB core has its own set of registers. USB0 Mentor Base Address is 0x4740_1400 while USB1 Mentor Core Registers Base address is 0x4740_1C00. Below offset of the core registers The absolute address is computed by summing the CORE_OFFSET with USBn_OFFSET.

15.5.6.1 Common USB Registers

These registers provide control and status for the complete core. Table 15-160 lists the registers.

Table 15-160. Common USB Registers

Core Address Offset	Common USB Registers
0h	Function Address Register
1h	Power Management Register
2h	Interrupt Register for Endpoint 0 Plus Transmit Endpoints 1 to 15
4h	Interrupt Register for Receive Endpoints 1 to 15
6h	Interrupt Enable Register for INTRTX
8h	Interrupt Enable Register for INTRRX
Ah	Interrupt Register for Common USB Interrupts
Bh	Interrupt Enable Register for INTRUSB
Ch	Frame Number Register
Eh	Index Register for Selecting the Endpoint Status and Control Registers
Fh	Register to Enable the USB 2.0 Test Modes

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.6.1.1 Function Address Register (USBn_FADDR)

The function address register (USBn_FADDR) is an 8-bit register that should be written with the 7-bit address of the peripheral part of the transaction.

Since USB cores are configured with multipoint support, this register only applies to operations carried out when the controller is in peripheral mode. In Host mode, this register is ignored.

The function address register is shown in Figure 15-147 and described in Table 15-161.

Figure 15-147. Function Address Register (USBn_FADDR)

7	6	0
Reserved	FUNCADDR	
R-0h	W-0-7Fh	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-161. Function Address Register (USBn_FADDR) Field Descriptions

Bits	Field Name	Description
7	Reserved	Reserved
6-0	FUNCADDR	7_bit address of the peripheral part of the transaction. When used in peripheral mode, this register should be written with the address received through a SET_ADDRESS command, which is then used for decoding the function address in subsequent token packets.

15.5.6.1.2 Power Management Register (USBn_POWER)

The power management register (USBn_POWER) is an 8-bit register that is used for controlling suspend and resume signaling, and some basic operational aspects of the USB core. This register is shown in Figure 15-148 and described in Table 15-162.

Figure 15-148. Power Management Register (USBn_POWER)

7	6	5	4	3	2	1	0
ISOUPDATE	SOFTCONN	HSEN	HSMODE	RESET	RESUME	SUSPENDM	ENSUSPM
R/W-0-1h	R/W-0-1h	R/W-0-1h	R/W-0-1h	R/W-0-1h	R/W-0-1h	R/W-0-1h	R/W-0-1h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-162. Power Management Register (USBn_POWER) Field Descriptions

Bits	Field Name	Description
7	ISOUPDATE	When set, the USB controller will wait for an SOF token from the time TxPktRdy is set before sending the packet. If an IN token is received before an SOF token, then a zero length data packet will be sent. Note: this is only valid in Peripheral Mode. This bit only affects endpoints performing Isochronous transfers.
6	SOFTCONN	If Soft Connect/Disconnect feature is enabled, then the USB D+/D-lines are enabled when this bit is set and tri-stated when this bit is cleared. Note: This is only valid in Peripheral Mode.
5	HSEN	When set, the USB controller negotiates for high-speed mode when the device is reset by the hub. If not set, the device will only operate in full-speed mode.
4	HSMODE	This bit is set when the USB controller has successfully negotiated for high-speed mode.
3	RESET	This bit is set when reset signaling is present on the bus. Note: this bit is Read/Write in Host Mode, but read-only in peripheral mode.
2	RESUME	Set to generate resume signaling when the controller is in suspend mode. The bit should be cleared after 10 ms (a maximum of 15 ms) to end resume signaling. In Host mode, this bit is also automatically set when resume signaling from the target is detected while the USB controller is suspended.
1	SUSPENDM	In Host mode, this bit should be set to enter suspend mode. In peripheral mode, this bit is set on entry into suspend mode. It is cleared when the interrupt register is read, or the RESUME bit is set.
0	ENSUSPM	Set to enable the SUSPENDM output.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.6.1.3 Interrupt Register for Endpoint 0 Plus Transmit Endpoints 1 to 15 (USBn_INTRTX)

The interrupt register for endpoint 0 plus transmit endpoints 1 to 15 (USBn_INTRTX) is a 16-bit read-only register that indicates which interrupts are currently active for endpoint 0 and the TX endpoints 1–15. Note also that all active interrupts are cleared when this register is read.

The interrupt register for endpoint 0 plus transmit endpoints 1 to 15 is shown in Figure 15-149 and described in Table 15-163.

Figure 15-149. Interrupt Register for Endpoint 0 Plus Transmit Endpoints 1 to 15 (USBn_INTRTX)

15	14	13	12	11	10	9	8
EP15TX	EP14TX	EP13TX	EP12TX	EP11TX	EP10TX	EP9TX	EP8TX
R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h
7	6	5	4	3	2	1	0
EP7TX	EP6TX	EP5TX	EP4TX	EP3TX	EP2TX	EP1TX	EP0
R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-163. Interrupt Register for Endpoint 0 Plus Transmit Endpoints 1 to 15 (USBn_INTRTX) Field Descriptions

Bits	Field Name	Description
15	EP15TX	Transmit endpoint 15 interrupt active
14	EP14TX	Transmit endpoint 14 interrupt active
:	:	:
4	EP4TX	Transmit endpoint 4 interrupt active
3	EP3TX	Transmit endpoint 3 interrupt active
2	EP2TX	Transmit endpoint 2 interrupt active
1	EP1TX	Transmit endpoint 1 interrupt active
0	EP0	Endpoint 0 (transmit or receive) interrupt active

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.6.1.4 Interrupt Register for Receive Endpoints 1 to 15 (INTRRX)

The interrupt register for receive endpoints 1 to 15 (INTRRX) is a 16-bit read-only register that indicates which of the interrupts for Rx Endpoints 1 – 15 are currently active. Note also that all active interrupts are cleared when this register is read.

The interrupt register for receive endpoints 1 to 15 is shown in Figure 15-150 and described in Table 15-164.

Figure 15-150. Interrupt Register for Receive Endpoints 1 to 15 (INTRRX)

15	14	13	12	11	10	9	8
EP15RX	EP14RX	EP13RX	EP12RX	EP11RX	EP10RX	EP9RX	EP8RX
R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h
7	6	5	4	3	2	1	0
EP7RX	EP6RX	EP5RX	EP4RX	EP3RX	EP2RX	EP1RX	Reserved
R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-164. Interrupt Register for Receive Endpoints 1 to 15 (INTRRX) Field Descriptions

Bits	Field Name	Description
15	EP15RX	Receive endpoint 15 interrupt active
14	EP14RX	Receive endpoint 14 interrupt active
:	:	:
4	EP4RX	Receive endpoint 4 interrupt active
3	EP3RX	Receive endpoint 3 interrupt active
2	EP2RX	Receive endpoint 2 interrupt active
1	EP1RX	Receive endpoint 1 interrupt active
0	Reserved	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.6.1.5 Interrupt Enable Register for INTRTX (INTRTXE)

The interrupt enable register for INTRTX (INTRTXE) is a 16-bit register that provides interrupt enable bits for the interrupts in IntrTx. On reset, the bits corresponding to endpoint 0 and the TX endpoints are set to 1. This register is shown in Figure 15-151 and described in Table 15-165.

Figure 15-151. Interrupt Enable Register for INTRTX (INTRTXE)

15	14	13	12	11	10	9	8
EP15TX	EP14TX	EP13TX	EP12TX	EP11TX	EP10TX	EP9TX	EP8TX
R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h
7	6	5	4	3	2	1	0
EP7TX	EP6TX	EP5TX	EP4TX	EP3TX	EP2TX	EP1TX	EP0
R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-165. Interrupt Enable Register for INTRTX (INTRTXE) Field Descriptions

Bits	Field Name	Description
15	EP15TX	Transmit endpoint 15 interrupt enable
14	EP14TX	Transmit endpoint 14 interrupt enable
:	:	:
4	EP4TX	Transmit endpoint 4 interrupt enable
3	EP3TX	Transmit endpoint 3 interrupt enable
2	EP2TX	Transmit endpoint 2 interrupt enable
1	EP1TX	Transmit endpoint 1 interrupt enable
0	EP0	Endpoint 0 interrupt active

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.6.1.6 Interrupt Enable Register for INTRRX (INTRRXE)

The interrupt enable register for INTRRX (INTRRXE) is a 16-bit register that provides interrupt enable bits for the interrupts in USBn_INTRRX. On reset, the bits corresponding to the Rx endpoints are set to 1. This register is shown in Figure 15-152 and described in Table 15-166.

Figure 15-152. Interrupt Enable Register for INTRRX (INTRRXE)

15	14	13	12	11	10	9	8
EP15RX	EP14RX	EP13RX	EP12RX	EP11RX	EP10RX	EP9RX	EP8RX
R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h
7	6	5	4	3	2	1	0
EP7RX	EP6RX	EP5RX	EP4RX	EP3RX	EP2RX	EP1RX	Reserved
R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-166. Interrupt Enable Register for INTRRX (INTRRXE) Field Descriptions

Bits	Field Name	Description
15	EP15RX	Receive endpoint 15 interrupt enable
14	EP14RX	Receive endpoint 14 interrupt enable
:	:	:
4	EP4RX	Receive endpoint 4 interrupt enable
3	EP3RX	Receive endpoint 3 interrupt enable
2	EP2RX	Receive endpoint 2 interrupt enable
1	EP1RX	Receive endpoint 1 interrupt enable
0	Reserved	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.6.1.7 Interrupt Register for Common USB Interrupts (USBn_INTRUSB)

The interrupt register for common USB interrupts (USBn_INTRUSB) is an 8-bit read-only register that indicates which USB interrupts are currently active. All active interrupts will be cleared when this register is read. This register is shown in Figure 15-153 and described in Table 15-167.

Figure 15-153. Interrupt Register for Common USB Interrupts (USBn_INTRUSB)

7	6	5	4	3	2	1	0
VBUSERR	SESSREQ	DISCON	CONN	SOF	RESET_BABBLE	RESUME	SUSPEND
R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-167. Interrupt Register for Common USB Interrupts (USBn_INTRUSB) Field Descriptions

Bits	Field Name	Description
7	VBUSERR	Set when VBus drops below the VBus valid threshold during a session. Only valid when the USB controller is 'A' device. All active interrupts will be cleared when this register is read.
6	SESSREQ	Set when session request signaling has been detected. Only valid when USB controller is 'A' device.
5	DISCON	Set in host mode when a device disconnect is detected. Set in peripheral mode when a session ends.
4	CONN	Set when a device connection is detected. Only valid in host mode.
3	SOF	Set when a new frame starts.
2	RESET_BABBLE	Set in peripheral mode when reset signaling is detected on the bus set in host mode when babble is detected.
1	RESUME	Set when resume signaling is detected on the bus while the USB controller is in suspend mode.
0	SUSPEND	Set when suspend signaling is detected on the bus only valid in peripheral mode.

15.5.6.1.8 Interrupt Enable Register for INTRUSB (USBn_INTRUSBE)

The interrupt enable register for INTRUSB (USBn_INTRUSBE) is an 8-bit register that provides interrupt enable bits for each of the interrupts in USBn_INTRUSB. This register is shown in Figure 15-154 and described in Table 15-168.

Figure 15-154. Interrupt Enable Register for INTRUSB (USBn_INTRUSBE)

7	6	5	4	3	2	1	0
VBUSERR	SESSREQ	DISCON	CONN	SOF	RESET_BABBLE	RESUME	SUSPEND
R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-168. Interrupt Enable Register for INTRUSB (USBn_INTRUSBE) Field Descriptions

Bits	Field Name	Description
7	VBUSERR	Vbus error interrupt enable
6	SESSREQ	Session request interrupt enable
5	DISCON	Disconnect interrupt enable
4	CONN	Connect interrupt enable
3	SOF	Start of frame interrupt enable
2	RESET_BABBLE	Reset interrupt enable
1	RESUME	Resume interrupt enable
0	SUSPEND	Suspend interrupt enable

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.6.1.9 Frame Number Register (USBn_FRAME)

The frame number register (USBn_FRAME) is a 16-bit read-only register that holds the last received frame number. This register is shown in Figure 15-155 and described in Table 15-169.

Figure 15-155. Frame Number Register (USBn_FRAME)

15–11	10–0
Reserved	FRAMENUMBER
R-0h	R-0-7FFh

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-169. Frame Number Register (USBn_FRAME) Field Descriptions

Bits	Field Name	Description
15-11	Reserved	Reserved
10-0	FRAMENUMBER	Last received frame number

15.5.6.1.10 Index Register for Selecting the Endpoint Status and Control Registers (USBn_INDEX)

Each TX endpoint and each Rx endpoint have their own set of control/status registers located between Core_Offset 0x0100 – 0x01FF. In addition one set of TX control/status and one set of Rx control/status registers appear at Core_Offset 0x0010 – 0x0019. This block of memory can be used as a proxy to access endpoint configuration registers. Index is a 4-bit register that determines which endpoint control/status registers are accessed.

Before accessing an endpoint's control/status registers at Core_Offset 0x0010 – 0x0019, the endpoint number should be written to the Index register to ensure that the correct control/status registers appear in the memory map.

The index register for selecting the endpoint status and control registers is shown in Figure 15-156 and described in Table 15-170.

Figure 15-156. Index Register for Selecting the Endpoint Status and Control Registers (USBn_INDEX)

7–4	3–0
Reserved	EPSEL
R-0h	R/W-0-Fh

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-170. Index Register for Selecting the Endpoint Status and Control Registers (USBn_INDEX) Field Descriptions

Bits	Field Name	Description
7-4	Reserved	Reserved
3-0	EPSEL	Each transmit endpoint and each receive endpoint have their own set of control/status registers. EPSEL determines which endpoint control/status registers are accessed. Before accessing an endpoint's control/status registers, the endpoint number should be written to the Index register to ensure that the correct control/status registers appear in the memory-map.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.6.1.11 Register to Enable the USB 2.0 Test Modes (USBn_TESTMODE)

The register to enable the USB 2.0 test modes (USBn_TESTMODE) is an 8-bit register that is primarily used to put the USB controller into one of the four test modes for high-speed operation described in the USB 2.0 specification – in response to a SET FEATURE: TESTMODE command. It is not used in normal operation.

The register to enable the USB 2.0 test modes is shown in Figure 15-157 and described in Table 15-171.

Figure 15-157. Register to Enable the USB 2.0 Test Modes (USBn_TESTMODE)

7	6	5	4	3	2	1	0
FORCE_HOST	FIFO_ACCESS	FORCE_FS	FORCE_HS	TEST_PACKET	TEST_K	TEST_J	TEST_SE0_NAK
R/W-0-1h	W-0-1h	R/W-0-1h	R/W-0-1h	R/W-0-1h	R/W-0-1h	R/W-0-1h	R/W-0-1h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-171. Register to Enable the USB 2.0 Test Modes (USBn_TESTMODE) Field Descriptions

Bits	Field Name	Description
7	FORCE_HOST	Set this bit to forcibly put the USB controller into Host mode when SESSION bit is set, regardless of whether it is connected to any peripheral. The controller remains in Host mode until the session bit is cleared, even if a device is disconnected. And if the FORCE_HOST but remains set, it re-enters Host mode next time the SESSION bit is set. The operating speed is determined using the FORCE_HS and FORCE_FS bits.
6	FIFO_ACCESS	Set this bit to transfer the packet in EP0 Tx FIFO to EP0 Receive FIFO. It is cleared automatically.
5	FORCE_FS	Set this bit to force the USB controller into full-speed mode when it receives a USB reset.
4	FORCE_HS	Set this bit to force the USB controller into high-speed mode when it receives a USB reset.
3	TEST_PACKET	Set this bit to enter the Test_Packet test mode. In this mode, the USB controller repetitively transmits a 53-byte test packet on the bus, the form of which is defined in the Universal Serial Bus Specification Revision 2.0. Note: The test packet has a fixed format and must be loaded into the endpoint 0 FIFO before the test mode is entered.
2	TEST_K	Set this bit to enter the Test_K test mode. In this mode, the USB controller transmits a continuous K on the bus.
1	TEST_J	Set this bit to enter the Test_J test mode. In this mode, the USB controller transmits a continuous J on the bus.
0	TEST_SE0_NAK	Set this bit to enter the Test_SE0_NAK test mode. In this mode, the USB controller remains in high-speed mode, but responds to any valid IN token with a NAK.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.6.2 Indexed Register Space

This is a block of the core register space that is used as a proxy to access the configuration and status registers of an endpoint. Mapping is done by programming the INDEX register with the endpoint number. Table 15-172 lists the registers.

Table 15-172. Indexed Region Registers

Core Address Offset	Indexed Region Registers
10h	Maximum Packet Size for Peripheral/Host Transmit Endpoint
12h	Control Status Register for Endpoint 0 in Peripheral Mode
12h	Control Status Register for Endpoint 0 in Host Mode
12h	Control Status Register for Peripheral Transmit Endpoint
12h	Control Status Register for Host Transmit Endpoint
14h	Maximum Packet Size for Peripheral/Host Receive Endpoint
16h	Control Status Register for Peripheral Receive Endpoint
16h	Control Status Register for Host Receive Endpoint
18h	Count 0 Register
18h	Receive Count Register
1Ah	Type Register (Host mode only)
1Ah	Transmit Type Register (Host mode only)
1Bh	NAKLimit0 Register (Host mode only)
1Bh	Transmit Interval Register (Host mode only)
1Ch	Receive Type Register (Host mode only)
1Dh	Receive Interval Register (Host mode only)
1Fh	Configuration Data Register

15.5.6.2.1 Maximum Packet Size for Peripheral/Host Transmit Endpoint Register (USBn_TXMAXP)

The maximum packet size for peripheral/host transmit endpoint register (USBn_TXMAXP) defines the maximum amount of data that can be transferred through the selected Tx endpoint in a single operation. There is a TxMaxP register for each Tx endpoint (except Endpoint 0). Endpoint 0 resource is hardened in design and max packet size is set to 64 bytes and no user configuration is required or does not exist.

The maximum packet size for peripheral/host transmit endpoint register is shown in Figure 15-158 and described in Table 15-173.

Figure 15-158. Maximum Packet Size for Peripheral/Host Transmit Endpoint Register (USBn_TXMAXP)

15 – 11	10 – 0
Reserved	MAXPAYLOAD
R-0h	R/W-0-400h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-173. Maximum Packet Size for Peripheral/Host Transmit Endpoint (USBn_TXMAXP) Field Descriptions

Bits	Field Name	Description
15-11	Reserved	Reserved
10-0	MAXPAYLOAD	The maximum payload transmitted in a single transaction. The value set can be up to 1024 bytes, but is subject to the constraints placed by the USB specification on packet sizes for bulk, interrupt, and isochronous transfers in full-speed and high-speed operations. The value written to this register should match the wMaxPacketSize field of the standard endpoint descriptor for the associated endpoint. A mismatch could cause unexpected results.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.6.2.2 Control Status Register for Endpoint 0 in Peripheral Mode (USBn_PERI_CSR0)

The control status register for endpoint 0 in peripheral mode (USBn_PERI_CSR0) is a 16-bit register that provides control and status bits for endpoint 0 when USB controller assumes the role of a peripheral. This register is shown in Figure 15-159 and described in Table 15-174.

Figure 15-159. Control Status Register for Endpoint 0 in Peripheral Mode (USBn_PERI_CSR0)

15						9	8
Reserved							FLUSHFIFO
R-0h							W-0-1h
7	6	5	4	3	2	1	0
SERV_ SETUPEND	SERV_ RXPKTRDY	SENDSTALL	SETUPEND	DATAEND	SENTSTALL	TXPKTRDY	RXPKTRDY
W-0-1h	W-0-1h	W-0-1h	R-0-1h	W-0-1h	R/W-0-1h	R/W-0-1h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-174. Control Status Register for Endpoint 0 in Peripheral Mode (USBn_PERI_CSR0) Field Descriptions

Bits	Field Name	Description
15-9	Reserved	Reserved
8	FLUSHFIFO	Set this bit to flush the next packet to be transmitted/read from the endpoint 0 FIFO. The FIFO pointer is reset and the TXPKTRDY/RXPKTRDY bit is cleared. Note: FLUSHFIFO has no effect unless TXPKTRDY/RXPKTRDY is set.
7	SERV_SETUPEND	Set this bit to clear the SETUPEND bit. It is cleared automatically.
6	SERV_RXPKTRDY	Set this bit to clear the RXPKTRDY bit. It is cleared automatically.
5	SENDSTALL	Set this bit to terminate the current transaction. The STALL handshake will be transmitted and then this bit will be cleared automatically.
4	SETUPEND	This bit will be set when a control transaction ends before the DATAEND bit has been set. An interrupt is generated, and the FIFO will be flushed at this time. The bit is cleared by the writing a 1 to the SERV_SETUPEND bit.
3	DATAEND	Set this bit to 1: a. When setting TXPKTRDY for the last data packet. b. When clearing RXPKTRDY after unloading the last data packet. c. When setting TXPKTRDY for a zero length data packet. It is cleared automatically.
2	SENTSTALL	This bit is set when a STALL handshake is transmitted. This bit should be cleared.
1	TXPKTRDY	Set this bit after loading a data packet into the FIFO. It is cleared automatically when the data packet has been transmitted. An interrupt is generated (if enabled) when the bit is cleared.
0	RXPKTRDY	This bit is set when a data packet has been received. An interrupt is generated when this bit is set. This bit is cleared by setting the SERV_RXPKTRDY bit.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.6.2.3 Control Status Register for Endpoint 0 in Host Mode (USBn_HOST_CSR0)

The control status register for endpoint 0 in host mode (USBn_HOST_CSR0) is a 16-bit register that provides control and status bits for endpoint 0 when USB controller assumes the role of a host. This register is shown in Figure 15-160 and described in Table 15-175.

Figure 15-160. Control Status Register for Endpoint 0 in Host Mode (USBn_HOST_CSR0)

15			12	11	10	9	8
Reserved				DSPING	DATATOGWREN	DATATOG	FLUSHFIFO
R-0h				R/W-0-1h	W-0-1h	R/W-0-1h	W-0-1h

7	6	5	4	3	2	1	0
NAK_TIMEOUT	STATUSPKT	REQPKT	ERROR	SETUPPKT	RXSTALL	TXPKTRDY	RXPKTRDY
W-0-1h	R/W-0-1h	R/W-0-1h	W-0-1h	R/W-0-1h	R/W-0-1h	R/W-0-1h	R/W-0-1h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-175. Control Status Register for Endpoint 0 in Host Mode (USBn_HOST_CSR0) Field Descriptions

Bits	Field Name	Description
15-12	Reserved	Reserved
11	DSPING	The CPU writes a 1 to the DSPING bit to instruct the core not to issue PING tokens in the data and status phases of a high-speed control transfer (for use with devices that do not respond to PING).
10	DATATOGWREN	Write 1 to this bit to enable the DATATOG bit to be written. This bit is automatically cleared once the new value is written to DATATOG.
9	DATATOG	When read, this bit indicates the current state of the EP0 data toggle. If DATATOGWREN is high, this bit can be written with the required setting of the data toggle. If DATATOGWREN is low, any value written to this bit is ignored.
8	FLUSHFIFO	Write 1 to this bit to flush the next packet to be transmitted/read from the Endpoint 0 FIFO. The FIFO pointer is reset and the TXPKTRDY/RXPKTRDY bit is cleared. Note: FLUSHFIFO has no effect unless TXPKTRDY/RXPKTRDY is set.
7	NAK_TIMEOUT	This bit will be set when endpoint 0 is halted following the receipt of NAK responses for longer than the time set by the NAKLIMIT0 register. This bit should be cleared to allow the endpoint to continue.
6	STATUSPKT	Set this bit at the same time as the TXPKTRDY or REQPKT bit is set, to perform a status stage transaction. Setting this bit ensures that the data toggle is set so that a DATA1 packet is used for the Status Stage transaction.
5	REQPKT	Set this bit to request an IN transaction. It is cleared when RXPKTRDY is set.
4	ERROR	This bit is set when three attempts have been made to perform a transaction with no response from the peripheral. You should clear this bit. An interrupt is generated when this bit is set.
3	SETUPPKT	Set this bit, at the same time as the TXPKTRDY bit is set, to send a SETUP token instead of an OUT token for the transaction.
2	RXSTALL	This bit is set when a STALL handshake is received. You should clear this bit.
1	TXPKTRDY	Set this bit after loading a data packet into the FIFO. It is cleared automatically when the data packet has been transmitted. An interrupt is generated (if enabled) when the bit is cleared.
0	RXPKTRDY	This bit is set when a data packet has been received. An interrupt is generated when this bit is set. Clear this bit by setting the SERV_RXPKTRDY bit.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.6.2.4 Control Status Register for Peripheral Transmit Endpoint (USBn_PERI_TXCSR)

The control status register for peripheral transmit endpoint (USBn_PERI_TXCSR) is a 16-bit register that provides control and status bits for transfers through the currently-selected Tx endpoint when controller assumes the role of a peripheral. There is a TXCSR register for each configured Tx endpoint (not including Endpoint 0).

The control status register for peripheral transmit endpoint is shown in Figure 15-161 and described in Table 15-176.

Figure 15-161. Control Status Register for Peripheral Transmit Endpoint (USBn_PERI_TXCSR)

15	14	13	12	11	10	9	8
AUTOSET	ISO	MODE	DMAEN	FRCDATATOG	DMAMODE	Reserved	
R/W-0-1h	R/W-0-1h	R/W-0-1h	R/W-0-1h	R/W-0-1h	R/W-0-1h	R-0h	

7	6	5	4	3	2	1	0
Reserved	CLRDATATOG	SENTSTALL	SENDSTALL	FLUSHFIFO	UNDERRUN	FIFONOTEMPTY	TXPKTRDY
R-0h	W-0-1h	R/W-0-1h	R/W-0-1h	R/W-0-1h	R/W-0-1h	R/W-0-1h	R/W-0-1h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-176. Control Status Register for Peripheral Transmit Endpoint (USBn_PERI_TXCSR) Field Descriptions

Bits	Field Name	Description
15	AUTOSET	DMA Mode: The CPU needs to set the AUTOSET bit prior to enabling the Tx DMA. CPU Mode: If the CPU sets the AUTOSET bit, the TXPKTRDY bit will be automatically set when data of the maximum packet size (value in TXMAXP) is loaded into the Tx FIFO. If a packet of less than the maximum packet size is loaded, then the TXPKTRDY bit will have to be set manually.
14	ISO	Set this bit to enable the Tx endpoint for Isochronous transfers, and clear it to enable the Tx endpoint for Bulk or Interrupt transfers.
13	MODE	Set this bit to enable the endpoint direction as Tx, and clear the bit to enable it as Rx. Note: This bit has any effect only where the same endpoint FIFO is used for both Transmit and Receive transactions.
12	DMAEN	Set this bit to enable the DMA request for the Tx endpoint.
11	FRCDATATOG	Set this bit to force the endpoint data toggle to switch and the data packet to be cleared from the FIFO, regardless of whether an ACK was received. This can be used by Interrupt Tx endpoints that are used to communicate rate feedback for Isochronous endpoints.
10	DMAMODE	This bit should always be set to 1 when the DMA is enabled.
9-7	Reserved	Reserved
6	CLRDATATOG	Write a 1 to this bit to reset the endpoint data toggle to 0.
5	SENTSTALL	This bit is set automatically when a STALL handshake is transmitted. The FIFO is flushed and the TXPKTRDY bit is cleared. You should clear this bit.
4	SENDSTALL	Write a 1 to this bit to issue a STALL handshake to an IN token. Clear this bit to terminate the stall condition. Note: This bit has no effect where the endpoint is being used for Isochronous transfers.
3	FLUSHFIFO	Write a 1 to this bit to flush the next packet to be transmitted from the endpoint Tx FIFO. The FIFO pointer is reset and the TXPKTRDY bit is cleared. Note: FlushFIFO has no effect unless the TXPKTRDY bit is set. Also note that, if the FIFO is double-buffered, FlushFIFO may need to be set twice to completely clear the FIFO.
2	UNDERRUN	This bit is set automatically if an IN token is received when TXPKTRDY is not set. You should clear this bit.
1	FIFONOTEMPTY	This bit is set when there is at least 1 packet in the Tx FIFO. You should clear this bit.
0	TXPKTRDY	Set this bit after loading a data packet into the FIFO. It is cleared automatically when a data packet has been transmitted. An interrupt is generated (if enabled) when the bit is cleared.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.6.2.5 Control Status Register for Host Transmit Endpoint (USBn_HOST_TXCSR)

The control status register for host transmit endpoint (USBn_HOST_TXCSR) is a 16-bit register that provides control and status bits for transfers through the currently-selected Tx endpoint when controller assumes the role of a host. There is a TXCSR register for each configured Tx endpoint (not including Endpoint 0).

The control status register for host transmit endpoint is shown in Figure 15-162 and described in Figure 15-162.

Figure 15-162. Control Status Register for Host Transmit Endpoint (USBn_HOST_TXCSR)

15	14	13	12	11	10	9	8
AUTOSET	Reserved	MODE	DMAEN	FRCDATATOG	DMAMODE	DATATOGWREN	DATATOG
R/W-0-1h	R-0h		R/W-0-1h		R/W-0-1h	W-0-1h	R/W-0-1h

7	6	5	4	3	2	1	0
NAK_TIMEOUT	CLRDATATOG	RXSTALL	SETUPPKT	FLUSHFIFO	ERROR	FIFONOTEMPTY	TXPKTRDY
R/W-0-1h	W-0-1h	R/W-0-1h	R/W-0-1h	W-0-1h	R/W-0-1h	R/W-0-1h	R/W-0-1h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-177. Control Status Register for Host Transmit Endpoint (USBn_HOST_TXCSR) Field Descriptions

Bits	Field Name	Description
15	AUTOSET	DMA Mode: The CPU needs to set the AUTOSET bit prior to enabling the Tx DMA. CPU Mode: If the CPU sets the AUTOSET bit, the TXPKTRDY bit will be automatically set when data of the maximum packet size (value in TXMAXP) is loaded into the Tx FIFO. If a packet of less than the maximum packet size is loaded, then the TXPKTRDY bit will have to be set manually.
14	Reserved	Reserved
13	MODE	Set this bit to enable the endpoint direction as Tx, and clear the bit to enable it as Rx. Note: This bit has any effect only where the same endpoint FIFO is used for both Transmit and Receive transactions.
12	DMAEN	Set this bit to enable the DMA request for the Tx endpoint.
11	FRCDATATOG	Set this bit to force the endpoint data toggle to switch and the data packet to be cleared from the FIFO, regardless of whether an ACK was received. This can be used by Interrupt Tx endpoints that are used to communicate rate feedback for Isochronous endpoints.
10	DMAMODE	This bit should always be set to 1 when the DMA is enabled.
9	DATATOGWREN	Write 1 to this bit to enable the DATATOG bit to be written. This bit is automatically cleared once the new value is written to DATATOG.
8	DATATOG	When read, this bit indicates the current state of the Tx EP data toggle. If DATATOGWREN is high, this bit can be written with the required setting of the data toggle. If DATATOGWREN is low, any value written to this bit is ignored.
7	NAK_TIMEOUT	This bit will be set when the Tx endpoint is halted following the receipt of NAK responses for longer than the time set as the NAKLIMIT by the TXINTERVAL register. It should be cleared to allow the endpoint to continue. Note: This is valid only for Bulk endpoints.
6	CLRDATATOG	Write a 1 to this bit to reset the endpoint data toggle to 0.
5	RXSTALL	This bit is set when a STALL handshake is received. The FIFO is flushed and the TXPKTRDY bit is cleared (see below). You should clear this bit.
4	SETUPPKT	Set this bit at the same time as TXPKTRDY is set, to send a SETUP token instead of an OUT token for the transaction. Note: Setting this bit also clears the DATATOG bit.
3	FLUSHFIFO	Write a 1 to this bit to flush the next packet to be transmitted from the endpoint Tx FIFO. The FIFO pointer is reset and the TXPKTRDY bit (below) is cleared. Note: FlushFIFO has no effect unless the TXPKTRDY bit is set. Also note that, if the FIFO is double-buffered, FLUSHFIFO may need to be set twice to completely clear the FIFO.
2	ERROR	The USB controller sets this bit when 3 attempts have been made to send a packet and no handshake packet has been received. You should clear this bit. An interrupt is generated when the bit is set. This is valid only when the endpoint is operating in Bulk or Interrupt mode.
1	FIFONOTEMPTY	The USB controller sets this bit when there is at least 1 packet in the Tx FIFO.
0	TXPKTRDY	Set this bit after loading a data packet into the FIFO. It is cleared automatically when a data packet has been transmitted. An interrupt is generated (if enabled) when the bit is cleared.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.6.2.6 Maximum Packet Size for Peripheral/Host Receive Endpoint Register (USBn_RXMAXP)

The maximum packet size for peripheral/host receive endpoint register (USBn_RXMAXP) defines the maximum amount of data that can be transferred through the selected Rx endpoint in a single operation. There is a RxMaxP register for each Rx endpoint (except Endpoint 0). Endpoint 0 resource is hardened in design and max packet size is set to 64 bytes and no user configuration is required or does not exist.

The maximum packet size for peripheral/host receive endpoint register is shown in Figure 15-163 and described in Table 15-178.

Figure 15-163. Maximum Packet Size for Peripheral/Host Receive Endpoint Register (USBn_RXMAXP)

15 – 11	10 – 0
Reserved	MAXPAYLOAD
R-0h	RW-0-400h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-178. Maximum Packet Size for Peripheral Host Receive Endpoint (USBn_RXMAXP) Field Descriptions

Bit	Field	Description
15-11	Reserved	Reserved
10-0	MAXPAYLOAD	Defines the maximum amount of data that can be transferred through the selected Receive endpoint in a single frame/microframe (high-speed transfers). The value set can be up to 1024 bytes, but is subject to the constraints placed by the USB specification on packet sizes for bulk, interrupt, and isochronous transfers in full-speed and high-speed operations. The value written to this register should match the wMaxPacketSize field of the standard endpoint descriptor for the associated endpoint. A mismatch could cause unexpected results.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.6.2.7 Control Status Register for Peripheral Receive Endpoint (USBn_PERI_RXCSR)

The control status register for peripheral receive endpoint (USBn_PERI_RXCSR) is a 16-bit register that provides control and status bits for transfers through the currently-selected Tx endpoint when controller assumes the role of a peripheral. There is a RXCSR register for each configured Rx endpoint (not including endpoint 0).

The control status register for peripheral receive endpoint is shown in Figure 15-164 and described in Table 15-179.

Figure 15-164. Control Status Register for Peripheral Receive Endpoint (USBn_PERI_RXCSR)

15	14	13	12
AUTOCLEAR	ISO	DMAEN	DISNYET
R/W-0-1h	R/W-0-1h	R/W-0-1h	R/W-0-1h
11	10	9	8
DMAMODE	DATATOGWREN	DATATOG	Reserved
R/W-0-1h	W-0-1h	R/W-0-1h	R-0h
7	6	5	4
CLRDATATOG	SENTSTALL	SENDSTALL	FLUSHFIFO
W-0-1h	R/W-0-1h	R/W-0-1h	W-0-1h
3	2	1	0
DATAERROR	OVERRUN	FIFOFULL	RXPKTRDY
R-0-1h	R/W-0-1h	R-0-1h	R/W-0-1h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-179. Control Status Register for Peripheral Receive Endpoint (USBn_PERI_RXCSR) Field Descriptions

Bit	Field	Description
15	AUTOCLEAR	DMA Mode: The CPU sets the AUTOCLEAR bit prior to enabling the Rx DMA. CPU Mode: If the CPU sets the AUTOCLEAR bit, then the RXPKTRDY bit will be automatically cleared when a packet of RXMAXP bytes has been unloaded from the Receive FIFO. When packets of less than the maximum packet size are unloaded, RXPKTRDY will have to be cleared manually.
14	ISO	Set this bit to enable the Receive endpoint for Isochronous transfers, and clear it to enable the Receive endpoint for bulk/interrupt transfers.
13	DMAEN	Set this bit to enable the DMA request for the Receive endpoints.
12	DISNYET	DISNYET: Applies only for bulk/interrupt transactions: The CPU sets this bit to disable the sending of NYET handshakes. When set, all successfully received Rx packets are ACK'd including at the point at which the FIFO becomes full. Note: This bit only has any effect in high-speed mode, in which mode it should be set for all Interrupt endpoints. PID_ERROR: Applies only for ISO Transactions: The core sets this bit to indicate a PID error in the received packet.
11	DMAMODE	Always clear this bit to 0.
10	DATATOGWREN	Write 1 to this bit to enable the DATATOG bit to be written. This bit is automatically cleared once the new value is written to DATATOG.
9	DATATOG	When read, this bit indicates the current state of the Receive EP data toggle. If DATATOGWREN is high, this bit can be written with the required setting of the data toggle. If DATATOGWREN is low, any value written to this bit is ignored.
8	Reserved	Reserved
7	CLRDATATOG	Write a 1 to this bit to reset the endpoint data toggle to 0.
6	SENTSTALL	This bit is set when a STALL handshake is transmitted. The FIFO is flushed and the TXPKTRDY bit is cleared. You should clear this bit.
5	SENDSTALL	Write a 1 to this bit to issue a STALL handshake. Clear this bit to terminate the stall condition. Note: This bit has no effect where the endpoint is being used for Isochronous transfers.
4	FLUSHFIFO	Write a 1 to this bit to flush the next packet to be read from the endpoint Receive FIFO. The FIFO pointer is reset and the RXPKTRDY bit is cleared. Note: FLUSHFIFO has no effect unless RXPKTRDY is set. Also note that, if the FIFO is double-buffered, FLUSHFIFO may need to be set twice to completely clear the FIFO.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-179. Control Status Register for Peripheral Receive Endpoint (USBn_PERI_RXCSR) Field Descriptions (continued)

Bit	Field	Description
3	DATAERROR	This bit is set when RXPKTRDY is set if the data packet has a CRC or bit-stuff error. It is cleared when RXPKTRDY is cleared. Note: This bit is only valid when the endpoint is operating in ISO mode. In Bulk mode, it always returns zero.
2	OVERRUN	This bit is set if an OUT packet cannot be loaded into the Receive FIFO. You should clear this bit. Note: This bit is only valid when the endpoint is operating in ISO mode. In bulk mode, it always returns zero.
1	FIFOFULL	This bit is set when no more packets can be loaded into the Receive FIFO.
0	RXPKTRDY	This bit is set when a data packet has been received. You should clear this bit when the packet has been unloaded from the Receive FIFO. An interrupt is generated when the bit is set.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.6.2.8 Control Status Register for Host Receive Endpoint (USBn_HOST_RXCSR)

The control status register for host receive endpoint (USBn_HOST_RXCSR) is a 16-bit register that provides control and status bits for transfers through the currently-selected Rx endpoint when controller assumes the role of a host. There is a TXCSR register for each configured Rx endpoint (not including endpoint 0).

The control status register for host receive endpoint is shown in Figure 15-165 and described in Table 15-180.

Figure 15-165. Control Status Register for Host Receive Endpoint (USBn_HOST_RXCSR)

15	14	13	12
AUTOCLEAR	AUTOREQ	DMAEN	DISNYET
R/W-0-1h	R/W-0-1h	R/W-0-1h	R/W-0-1h
11	10	9	8
DMAMODE	DATATOGWREN	DATATOG	Reserved
R/W-0-1h	W-0-1h	R/W-0-1h	R-0h
7	6	5	4
CLRDATATOG	RXSTALL	REQPKT	FLUSHFIFO
W-0-1h	R/W-0-1h	R/W-0-1h	W-0-1h
3	2	1	0
DATAERR_NAKTIMEOUT	ERROR	FIFOFULL	RXPKTRDY
R-0-1h	R/W-0-1h	R-0-1h	R/W-0-1h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-180. Control Status Register for Host Receive Endpoint (USBn_HOST_RXCSR) Field Descriptions

Bit	Field	Description
15	AUTOCLEAR	DMA Mode: The CPU sets the AUTOCLEAR bit prior to enabling the Rx DMA. CPU Mode: If the CPU sets the AUTOCLEAR bit, then the RXPKTRDY bit will be automatically cleared when a packet of RXMAXP bytes has been unloaded from the Receive FIFO. When packets of less than the maximum packet size are unloaded, RXPKTRDY will have to be cleared manually.
14	AUTOREQ	If the CPU sets the AUTOREQ bit, then the REQPKT bit will be automatically set when the RXPKTRDY bit is cleared. Note: This bit is automatically cleared when a short packet is received.
13	DMAEN	Set this bit to enable the DMA request for the Receive endpoints.
12	DISNYET	Set this bit to disable the sending of NYET handshakes. When set, all successfully received Receive packets are ACKED including at the point at which the FIFO becomes full. Note: This bit only has any effect in high-speed mode, in which mode it should be set for all Interrupt endpoints.
11	DMAMODE	Always clear this bit to 0.
10	DATATOGWREN	Write 1 to this bit to enable the DATATOG bit to be written. This bit is automatically cleared once the new value is written to DATATOG.
9	DATATOG	When read, this bit indicates the current state of the Receive EP data toggle. If DATATOGWREN is high, this bit can be written with the required setting of the data toggle. If DATATOGWREN is low, any value written to this bit is ignored.
8	Reserved	Reserved
7	CLRDATATOG	Write a 1 to this bit to reset the endpoint data toggle to 0.
6	RXSTALL	When a STALL handshake is received, this bit is set and an interrupt is generated. You should clear this bit.
5	REQPKT	Write a 1 to this bit to request an IN transaction. It is cleared when RXPKTRDY is set.
4	FLUSHFIFO	Write a 1 to this bit to flush the next packet to be read from the endpoint Receive FIFO. The FIFO pointer is reset and the RXPKTRDY bit is cleared. Note: FLUSHFIFO has no effect unless RXPKTRDY is set. Also note that, if the FIFO is double-buffered, FLUSHFIFO may need to be set twice to completely clear the FIFO.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 15-180. Control Status Register for Host Receive Endpoint (USBn_HOST_RXCSR) Field Descriptions (continued)

Bit	Field	Description
3	DATAERR_NAKTIMEOUT	When operating in ISO mode, this bit is set when RXPKTRDY is set if the data packet has a CRC or bit-stuff error and cleared when RXPKTRDY is cleared. In Bulk mode, this bit will be set when the Receive endpoint is halted following the receipt of NAK responses for longer than the time set as the NAK Limit by the RXINTERVAL register. You should clear this bit to allow the endpoint to continue.
2	ERROR	The USB controller sets this bit when three attempts have been made to receive a packet and no data packet has been received. You should clear this bit. An interrupt is generated when the bit is set. Note: This bit is only valid when the transmit endpoint is operating in Bulk or Interrupt mode. In ISO mode, it always returns zero.
1	FIFOFULL	This bit is set when no more packets can be loaded into the Receive FIFO.
0	RXPKTRDY	This bit is set when a data packet has been received. You should clear this bit when the packet has been unloaded from the Receive FIFO. An interrupt is generated when the bit is set.

15.5.6.2.9 Count 0 Register (USBn_COUNT0)

The count 0 register (USBn_COUNT0) is a 7-bit read-only register that indicates the number of received data bytes in the Endpoint 0 FIFO. The value returned changes as the contents of the FIFO change and is only valid while RxPktRdy (bit 0) is set. This register is shown in Figure 15-166 and described in Table 15-181.

Figure 15-166. Count 0 Register (USBn_COUNT0)

15 – 7	6 – 0
Reserved	EP0RXCOUNT
R-0h	R-0-7h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-181. Count 0 Register (USBn_COUNT0) Field Descriptions

Bit	Field	Description
15-7	Reserved	Reserved
6-0	EP0RXCOUNT	Indicates the number of received data bytes in the endpoint 0 FIFO. The value returned changes as the contents of the FIFO change and is only valid while RXPKTRDY of USBn_PERI_CSR0 or USBn_HOST_CSR0 is set.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.6.2.10 Receive Count Register (USBn_RXCOUNT)

The receive count register (USBn_RXCOUNT) is a 13-bit read-only register that holds the number of received data bytes in the packet currently in line to be read from the Rx FIFO for endpoints other than endpoint 0.

Note: The value returned changes as the FIFO is unloaded and is only valid while RxPktRdy (bit 0) is set.

The receive count register is shown in Figure 15-167 and described in Table 15-182.

Figure 15-167. Receive Count Register (USBn_RXCOUNT)

15–13	12–0
Reserved	EPRXCOUNT
R-0h	R-0-1FFFh

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-182. Receive Count Register (USBn_RXCOUNT) Field Descriptions

Bit	Field	Description
15-13	Reserved	Reserved
12-0	EPRXCOUNT	Holds the number of received data bytes in the packet in the Receive FIFO. The value returned changes as the contents of the FIFO change and is only valid while RXPKTRDY of PERI_RXCSR or HOST_RXCSR is set.

15.5.6.2.11 Type Register (Host Mode Only) (USBn_HOST_TYPE0)

The type register (Host mode only) (USBn_HOST_TYPE0) is an 8-bit register, meaningful only when controller assumes the role of a host, of which only bits 6 and 7 are implemented. These bits should be written with the operating speed of the targeted device. This register is shown in Figure 15-168 and described in Table 15-183.

Figure 15-168. Type Register (Host Mode Only) (USBn_HOST_TYPE0)

7–6	5–0
SPEED	Reserved
R/W-0-3h	R-0h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-183. Type Register (Host Mode Only) (USBn_HOST_TYPE0) Field Descriptions

Bit	Field	Value	Description
7-6	SPEED		Operating Speed of Target Device
		0h	Illegal
		1h	High
		2h	Full
		3h	Low
5-0	Reserved	0	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.6.2.12 Transmit Type Register (Host Mode Only) (USBn_HOST_TXTYPE)

The transmit type register (Host mode only) (USBn_HOST_TXTYPE) is an 8-bit register that should be written with the endpoint number to be targeted by the endpoint, the transaction protocol to use for the currently-selected Tx endpoint, and its operating speed. There is a TxType register for each configured Tx endpoint (except Endpoint 0).

The transmit type register is shown in Figure 15-169 and described in Table 15-184.

Figure 15-169. Transmit Type Register (Host Mode Only) (USBn_HOST_TXTYPE)

7 6	5 4	3 0
SPEED	PROT	TENDPN
R/W-0-3h	R/W-0-3h	R/W-0-Fh

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-184. Transmit Type Register (Host Mode Only) (USBn_HOST_TXTYPE) Field Descriptions

Bit	Field	Value	Description
7-6	SPEED		Operating Speed of Target Device
		0h	Illegal
		1h	High
		2h	Full
		3h	Low
5-4	PROT		Set this to select the required protocol for the transmit endpoint
		0h	Control
		1h	Isochronous
		2h	Bulk
		3h	Interrupt
3-0	TENDPN		Set this value to the endpoint number contained in the transmit endpoint descriptor returned to the USB controller during device enumeration.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.6.2.13 NAKLimit0 Register (Host Mode Only) (USBn_HOST_NAKLIMIT0)

The NAKLimit0 register (Host mode only) (USBn_HOST_NAKLIMIT0) is a 5-bit register that sets the number of frames/microframes (high-speed transfers) after which endpoint 0 should timeout on receiving a stream of NAK responses. (Equivalent settings for other endpoints can be made through their TxInterval and RxInterval registers.) The number of frames/microframes selected is 2(m-1) (where m is the value set in the register, valid values 2 – 16). If the host receives NAK responses from the target for more frames than the number represented by the Limit set in this register, the endpoint will be halted. Note: A value of 0 or 1 disables the NAK timeout function.

The NAKLimit0 register is shown in Figure 15-170 and described in Table 15-185.

Figure 15-170. NAKLimit0 Register (Host Mode Only) (USBn_HOST_NAKLIMIT0)

7 – 5	4 – 0
Reserved	EP0NAKLIMIT
R-0h	R/W-0-1Fh

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-185. NAKLimit0 Register (Host Mode Only) (USBn_HOST_NAKLIMIT0) Field Descriptions

Bit	Field	Description
7-5	Reserved	Reserved
4-0	EP0NAKLIMIT	Sets the number of frames/microframes (high-speed transfers) after which endpoint 0 should time out on receiving a stream of NAK responses. The number of frames/microframes selected is 2(-1) (where m is the value set in the register, valid values 2-16). If the host receives NAK responses from the target for more frames than the number represented by the Limit set in this register, the endpoint will be halted. Note: A value of 0 or 1 disables the NAK timeout function.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.6.2.14 Transmit Interval Register (Host Mode Only) (USBn_HOST_TXINTERVAL)

The transmit interval register (Host mode only) (USBn_HOST_TXINTERVAL) is an 8-bit register (meaningful only when controller assumes the role of a host) that, for interrupt and isochronous transfers, defines the polling interval for the currently-selected Tx endpoint. For bulk endpoints, this register sets the number of frames/microframes after which the endpoint should timeout on receiving a stream of NAK responses. There is a TxInterval register for each configured Tx endpoint (except endpoint 0).

The transmit interval register is shown in Figure 15-171 and described in Table 15-186.

Figure 15-171. Transmit Interval Register (Host Mode Only) (USBn_HOST_TXINTERVAL)

7							0
POLINTVL_NAKLIMIT							
R/W-0-FFh							

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-186. Transmit Interval Register (Host Mode Only) (USBn_HOST_TXINTERVAL) Field Descriptions

Bit	Field	Description			
7-0	POLINTVL_NAKLIMIT	For interrupt and isochronous transfers, defines the polling interval for the currently-selected transmit endpoint. For bulk endpoints, sets the number of frames/microframes after which the endpoint should timeout on receiving a stream of NAK responses. There is a transmit interval register for each configured transmit endpoint (except endpoint 0). In each case, the value that is set defines a number of frames/microframes (high-speed transfers), as follows:			
		Transfer Type	Speed	Valid values (m)	Interpretation
		Interrupt	Low Speed or Full Speed	1-255	Polling interval is m frames
				1-16	Polling interval is 2(-1) microframes
		Isochronous	Full Speed or High Speed	1-16	Polling interval is 2(-1) frames/microframes
		Bulk	Full Speed or High Speed	2-16	NAK Limit is 2(-1)frames/microframes
		Note: A value of 0 or 1 disables the NAK timeout function.			

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.6.2.15 Receive Type Register (Host Mode Only) (USBn_HOST_RXTYPE)

The receive type register (Host mode only) (USBn_HOST_RXTYPE) is an 8-bit register that should be written with the endpoint number to be targeted by the endpoint, the transaction protocol to use for the currently-selected Rx endpoint, and its operating speed. There is a RxType register for each configured Rx endpoint (except endpoint 0).

The receive type register is shown in Figure 15-172 and described in Table 15-187.

Figure 15-172. Receive Type Register (Host Mode Only) (USBn_HOST_RXTYPE)

7	6	5	4	3	0
SPEED		PROT		RENDPN	
R/W-0-3h		R/W-0-3h		R/W-0-Fh	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-187. Receive Type Register (Host Mode Only) (USBn_HOST_RXTYPE) Field Descriptions

Bits	Field Name	Value	Description
7-6	SPEED		Operating Speed of Target Device
		0h	Illegal
		1h	High
		2h	Full
		3h	Low
5-4	PROT		Set this to select the required protocol for the transmit endpoint
		0h	Control
		1h	Isochronous
		2h	Bulk
		3h	Interrupt
3-0	RENDPN		Set this value to the endpoint number contained in the Receive endpoint descriptor returned to the USB controller during device enumeration

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.6.2.16 Receive Interval Register (Host Mode Only) (USBn_HOST_RXINTERVAL)

The receive interval register (Host mode only) (USBn_HOST_RXINTERVAL) is an 8-bit register (only meaningful in host mode only) that, for Interrupt and Isochronous transfers, defines the polling interval for the currently-selected Rx endpoint. For bulk endpoints, this register sets the number of frames/microframes after which the endpoint should timeout on receiving a stream of NAK responses. There is a RxInterval register for each configured Rx endpoint (except endpoint 0).

The receive interval register is shown in Figure 15-173 and described in Table 15-188.

Figure 15-173. Receive Interval Register (Host Mode Only) (USBn_HOST_RXINTERVAL)

7							0
POLINTVL_NAKLIMIT							
R/W-0-FFh							

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-188. Transmit Interval Register (Host Mode Only) (USBn_HOST_RXINTERVAL) Field Descriptions

Bit	Field	Description			
7-0	POLINTVL_NAKLIMIT	For interrupt and isochronous transfers, defines the polling interval for the currently-selected transmit endpoint. For bulk endpoints, sets the number of frames/microframes after which the endpoint should timeout on receiving a stream of NAK responses. There is a transmit interval register for each configured transmit endpoint (except endpoint 0). In each case, the value that is set defines a number of frames/microframes (high-speed transfers), as follows:			
		Transfer Type	Speed	Valid values (m)	Interpretation
		Interrupt	Low Speed or Full Speed	1-255	Polling interval is m frames
			High Speed	1-16	Polling interval is 2(-1) microframes
		Isochronous	Full Speed or High Speed	1-16	Polling interval is 2(-1) frames/microframes
		Bulk	Full Speed or High Speed	2-16	NAK Limit is 2(-1)frames/microframes
		Note: A value of 0 or 1 disables the NAK timeout function.			

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.6.2.17 Configuration Data Register (USBn_CONFIGDATA)

The configuration data register (USBn_CONFIGDATA) is an 8-bit read-only register that returns information about the selected core configuration. This register information can only be accessed from Indexed Region and INDEX has to be programmed, with zero, selecting endpoint 0. This register is shown in Figure 15-174 and described in Table 15-189.

Figure 15-174. Configuration Data Register (USBn_CONFIGDATA)

7	6	5	4	3	2	1	0
MPRXE	MPTXE	BIGENDIAN	HBRXE	HBTXE	DYNFIFO	SOFTCONE	UTMIDATAWIDTH
R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h	R-0-1h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-189. Configuration Data Register (USBn_CONFIGDATA) Field Descriptions

Bit	Field	Value	Description
7	MPRXE		Indicates automatic amalgamation of bulk packets. (INDEX register needs to be set to zero.)
		0	Automatic amalgamation of bulk packets is not selected.
		1	Automatic amalgamation of bulk packets is selected.
6	MPTXE		Indicates automatic splitting of bulk packets. (INDEX register needs to be set to zero.)
		0	Automatic splitting of bulk packets is not selected.
		1	Automatic splitting of bulk packets is selected.
5	BIGENDIAN		Indicates endian ordering. (INDEX register needs to be set to zero.)
		0	Little-endian ordering is selected.
		1	Big-endian ordering is selected.
4	HBRXE		Indicates high-bandwidth Rx ISO endpoint support. (INDEX register needs to be set to zero.)
		0	High-bandwidth Rx ISO endpoint support is not selected.
		1	High-bandwidth Rx ISO endpoint support is selected.
3	HBTXE		Indicates high-bandwidth Tx ISO endpoint support. (INDEX register needs to be set to zero.)
		0	High-bandwidth Tx ISO endpoint support is not selected.
		1	High-bandwidth Tx ISO endpoint support is selected.
2	DYNFIFO		Indicates dynamic FIFO sizing. (INDEX register needs to be set to zero.)
		0	Dynamic FIFO sizing option is not selected.
		1	Dynamic FIFO sizing option is selected.
1	SOFTCONE		Indicates soft connect/disconnect. (INDEX register needs to be set to zero.)
		0	Soft connect/disconnect option is not selected.
		1	Soft connect/disconnect option is selected.
0	UTMIDATAWIDTH		Indicates selected UTMI data width. (INDEX register needs to be set to zero.)
		0	8 bits
		1	16 bits

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.7 FIFOs

This address range provides access to the endpoint FIFOs. Register offset addresses for this block is 20h to 5Fh.

15.5.7.1 Transmit and Receive FIFO Register for Endpoint 0 - 15 (USBn_FIFO0 – USBn_FIFO15)

This address range provides 16 addresses for CPU access to the FIFOs for each endpoint. Writing to these addresses loads data into the TxFIFO for the corresponding endpoint. Reading from these addresses unloads data from the RxFIFO for the corresponding endpoint. The address range is 20h – 5Fh and the FIFOs are located on 32-bit double-word boundaries (endpoint 0 at 20h, endpoint 1 at 24h ... Endpoint 15 at 5Ch).

Note 1: Transfers to and from FIFOs may be 8-bit, 16-bit or 32-bit as required, and any combination of access is allowed provided the data accessed is contiguous. However, all the transfers associated with one packet must be of the same width so that the data is consistently byte-, word- or double-word-aligned. The last transfer may however contain fewer bytes than the previous transfers in order to complete an odd-byte or odd-word transfer.

Note 2: Depending on the size of the FIFO and the expected maximum packet size, the FIFOs support either single-packet or doublepacket buffering. However, burst writing of multiple packets is not supported as flags need to be set after each packet is written.

Note 3: Following a STALL response or a Tx strike out error on dndpoint 1 – 15, the associated FIFO is completely flushed.

The transmit and receive FIFO register for endpoint 0 - 15 (USBn_FIFO0 – USBn_FIFO15) is shown in Figure 15-175 and described in Table 15-190.

Figure 15-175. Transmit and Receive FIFO Register for Endpoint 0 - 15 (USBn_FIFO0 – USBn_FIFO15)

31	0
DATA	
R/W-0-FFFF FFFFh	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-190. FIFOs: Transmit and Receive FIFO Register for Endpoint 0 - 15 (USBn_FIFO0 – USBn_FIFO15) Field Descriptions

Bit	Field	Description
31-0	DATA	Writing to these addresses loads data into the Transmit FIFO for the corresponding endpoint. Reading from these addresses unloads data from the Receive FIFO for the corresponding endpoint.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.7.2 Additional Control and Configuration Registers

Table 15-191 lists the additional Control and Configuration registers.

Table 15-191. Additional Control and Configuration Registers

Core Address Offset	Control and Configuration Registers
60h	Device Control Register
62h	Transmit Endpoint FIFO Size Register
63h	Receive Endpoint FIFO Size Register
64h	Transmit Endpoint FIFO Address Register
66h	Receive Endpoint FIFO Address Register
6Ch	Hardware Version Register

15.5.7.2.1 Device Control Register (USBn_DEVCTL)

The device control register (USBn_DEVCTL) is an 8-bit register that is used to select whether the USB controller is operating in peripheral mode or in host mode, and for controlling and monitoring the USB VBus line. If the PHY is suspended no PHY clock is received and the VBus is not sampled. This register is shown in Figure 15-176 and described in Table 15-192.

Figure 15-176. Device Control Register (USBn_DEVCTL)

7	6	5	4	3	2	1	0
BDEVICE	FSDEV	LSDEV	VBUS		HOSTMODE	HOSTREQ	SESSION

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-192. Device Control Register (USBn_DEVCTL) Field Descriptions

Bit	Field	Value	Description
7	BDEVICE		This read-only bit indicates whether the USB controller is operating as the 'A' device or the 'B' device.
		0	A device
		1	B device
6	FSDEV		This read-only bit is set when a full-speed or high-speed device has been detected being connected to the port (high-speed devices are distinguished from full-speed by checking for high-speed chirps when the device is reset). Only valid in Host mode.
5	LSDEV		This read-only bit is set when a low-speed device has been detected being connected to the port. Only valid in Host mode.
4-3	VBUS		These read-only bits encode the current VBus level as follows:
		0h	Below session end
		1h	Above session dnd, below AValid
		2h	Above AValid, below VBusValid
		3h	Above VBusValid
2	HOSTMODE		This read-only bit is set when the USB controller is acting as a Host.
1	HOSTREQ		When set, the USB controller initiates the Host negotiation when suspend mode is entered. It is cleared when Host negotiation is completed. ('B' device only)
0	SESSION		When operating as an 'A' device, you must set or clear this bit start or end a session. When operating as a 'B' device, this bit is set/cleared by the USB controller when a session starts/ends. You must also set this bit to initiate the session request protocol. When the USB controller is in suspend mode, you may clear the bit to perform a software disconnect. A special software routine is required to perform SRP. Details will be made available in a later document version.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.7.2.2 Transmit Endpoint FIFO Size Register (USBn_TXFIFOSZ)

The allocation of FIFO space to the different endpoints requires the specification for each Tx endpoint:

- The start address of the FIFO within the RAM block
- The maximum size of packet to be supported
- Whether double-buffering is required

The transmit endpoint FIFO size register (USBn_TXFIFOSZ) defines the amount of space that needs to be allocated to the FIFO in either single or double buffering context.

For more information, see the *Dynamic FIFO Allocation* section.

The transmit endpoint FIFO size register is shown in Figure 15-177 and described in Table 15-193.

Figure 15-177. Transmit Endpoint FIFO Size Register (USBn_TXFIFOSZ)

7	6	5	4	3	0
Reserved			DPB	SZ	
R-0h			R-0-1h	R-0-Fh	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-193. Transmit Endpoint FIFO Size Register (USBn_TXFIFOSZ) Field Descriptions

Bit	Field	Value	Description
7-5	Reserved		Reserved
4	DPB		Double packet buffering enable.
		0	Single packet buffering is supported.
		1	Double packet buffering is enabled
3-0	SZ		Maximum packet size to be allowed (before any splitting within the FIFO of Bulk packets prior to transmission). If m = SZ, the FIFO size is calculated as 2(m+3) for single packet buffering and 2(m+4) for dual packet buffering.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.7.2.3 Receive Endpoint FIFO Size Register (USBn_RXFIFOSZ)

The allocation of FIFO space to the different endpoints requires the specification for each Rx endpoint

- The start address of the FIFO within the RAM block
- The maximum size of packet to be supported
- Whether double-buffering is required

The receive endpoint FIFO size register (USBn_RXFIFOSZ) defines the amount of space that needs to be allocated to the FIFO in either single or double buffering context.

For more information, see the *Dynamic FIFO Allocation* section.

The receive endpoint FIFO size register is shown in Figure 15-178 and described in Table 15-194.

Figure 15-178. Receive Endpoint FIFO Size Register (USBn_RXFIFOSZ)

7	6	5	4	3	0
Reserved			DPB	SZ	
0h			0-1h	0-Fh	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-194. Receive Endpoint FIFO Size Register (USBn_RXFIFOSZ) Field Descriptions

Bit	Field	Value	Description
7-5	Reserved		Reserved
4	DPB		Double packet buffering enable.
		0	Single packet buffering is supported.
		1	Double packet buffering is enabled
3-0	SZ		Maximum packet size to be allowed (before any splitting within the FIFO of Bulk packets prior to transmission). If m = SZ, the FIFO size is calculated as 2(m+3) for single packet buffering and 2(m+4) for dual packet buffering.

15.5.7.2.4 Transmit Endpoint FIFO Address Register (USBn_TXFIFOADDR)

The allocation of FIFO space to the different endpoints requires the specification for each Tx endpoint:

- The start address of the FIFO within the RAM block
- The maximum size of packet to be supported
- Whether double-buffering is required

The transmit endpoint FIFO address register (USBn_TXFIFOADDR) the coding for the start address of the FIFO. Note that the 1st 64 bytes of the FIFO RAM is reserved for the use of endpoint 0. Allocation of FIFO should exclude this reserved location.

The transmit endpoint FIFO address register is shown in Figure 15-179 and described in Table 15-195.

Figure 15-179. Transmit Endpoint FIFO Address Register (USBn_TXFIFOADDR)

15	13	12	0
Reserved		ADDR	
0h		0-1FFFh	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-195. Transmit Endpoint FIFO Address Register (USBn_TXFIFOADDR) Field Descriptions

Bit	Field	Description
15-13	Reserved	Reserved
12-0	ADDR	Start address of endpoint FIFO in units of 8 bytes If m = ADDR, then the start address is 8 × m

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.7.2.5 Receive Endpoint FIFO Address Register (USBn_RXFIFOADDR)

The allocation of FIFO space to the different endpoints requires the specification for each Rx endpoint:

- The start address of the FIFO within the RAM block
- The maximum size of packet to be supported
- Whether double-buffering is required

The receive endpoint FIFO address register (USBn_RXFIFOADDR) the coding for the start address of the FIFO. Note that the 1st 64 bytes of the FIFO RAM is reserved for the use of endpoint 0. Allocation of FIFO should exclude this reserved location.

The receive endpoint FIFO address register is shown in Figure 15-180 and described in Table 15-196.

Figure 15-180. Receive Endpoint FIFO Address Register (USBn_RXFIFOADDR)

15–13	12–0
Reserved	ADDR
0h	0-1FFFh

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-196. Receive Endpoint FIFO Address Register (USBn_RXFIFOADDR) Field Descriptions

Bit	Field	Description
15-13	Reserved	Reserved
12-0	ADDR	Start address of endpoint FIFO in units of 8 bytes If m = ADDR, then the start address is 8 × m

15.5.7.2.6 Hardware Version Register (USBn_HWVERS)

The hardware version register (USBn_HWVERS) is a 16-bit read-only register that returns information about the version of the RTL from which the core hardware was generated, in particular the RTL version number (REVMAJ.REVMIN). This register is shown in Figure 15-181 and described in Table 15-197.

Figure 15-181. Hardware Version Register (USBn_HWVERS)

15	14–10	9–0
RC	REVMAJ	REVMIN
0-1h	0-1Fh	0-3E7h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-197. Hardware Version Register (USBn_HWVERS) Field Descriptions

Bit	Field	Description
15	RC	Set to 1 if RTL is used from a Release Candidate, rather than from a full release of the core.
14-10	REVMAJ	Major version of RTL. Range is 0-31.
9-0	REVMIN	Minor version of RTL. Range is 0-999.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.7.3 Target Endpoint Control Registers

Table 15-198 lists the target endpoint control registers. These registers provide target function and hub address details for each of the endpoints. Register addresses consumed by this block is 0x0080 to 0x00FF.

Table 15-198. Target Endpoint Control Registers

Core Address Offset	Target Endpoint Control Registers
80h	Transmit Function Address Register
82h	Transmit Hub Address Register
83h	Transmit Hub Port Register
84h	Receive Function Address Register
86h	Receive Hub Address Register
87h	Receive Hub Port Register

15.5.7.3.1 Transmit Function Address Register (USBn_TXFUNCADDRm)

The transmit function address register (USBn_TXFUNCADDRm) is a 7-bit read/write register (meaningful in host mode operation only) that records the address of the target function that is to be accessed through the associated endpoint (EPn). USBn_ TXFUNCADDR needs to be defined for each Tx endpoint that is used.

Note: USBn_TXFUNCADDRm needs to be defined for endpoint 0. This register, together with the USBn_TXHUBADDRm and USBn_TXHUBPORTm registers described below, allow the allocation of multiple devices to the USB controller endpoints.

The transmit function address register is shown in Figure 15-182 and described in Table 15-199.

Figure 15-182. Transmit Function Address Register (USBn_TXFUNCADDRm)

7	6 … 0
Reserved	FUNCADDR
0h	0-7Fh

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-199. Transmit Function Address (USBn_TXFUNCADDRm) Field Descriptions

Bit	Field	Description
7	Reserved	Reserved
6-0	FUNCADDR	Address of target function

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.7.3.2 Transmit Hub Address Register (USBn_TXHUBADDRm)

The transmit hub address register (USBn_TXHUBADDRm) is an 8-bit read/write register which, like USBn_TXHUBPORTm, only need to be written where a full- or low-speed device is connected to Tx endpoint EPn via a high-speed USB 2.0 hub that carries out the necessary transaction translation to convert between high-speed transmission and full-/low-speed transmission. In such circumstances:

- The lower 7 bits should record the address of this USB 2.0 hub
- The top bit should record whether the hub has multiple transaction translators (set to ‘0’ if single transaction translator; set to ‘1’ if multiple transaction translators).

Note: If endpoint 0 is connected to a hub, then USBn_TXHUBADDRm need to be defined for this endpoint.

The transmit hub address register is shown in Figure 15-183 and described in Table 15-200.

Figure 15-183. Transmit Hub Address Register (USBn_TXHUBADDRm)

7	6 … 0
Reserved	FUNCADDR
0h	0-7Fh

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-200. Transmit Hub Address Register (USBn_TXHUBADDRm) Field Descriptions

Bit	Field	Description
7	MULT_TRANS	Set to 1 if hub has multiple transaction translators. Cleared to 0 if only single transaction translator is available.
6-0	HUBADDR	Address of hub

15.5.7.3.3 Transmit Hub Port Register (USBn_TXHUBPORTm)

The transmit hub port register (USBn_TXHUBPORTm) (meaningful in host mode operation only) only needs to be written where a full- or low-speed device is connected to Tx Endpoint EPn via a high-speed USB 2.0 hub which carries out the necessary transaction translation. In such circumstances, these 7-bit read/write registers need to be used to record the port of that USB 2.0 hub through which the target associated with the endpoint is accessed.

Note: If Endpoint 0 is connected to a hub, then USBn_TXHUBPORT needs to be defined for this endpoint.

The transmit hub port register is shown in Figure 15-184 and described in Table 15-201.

Figure 15-184. Transmit Hub Port Register (USBn_TXHUBPORTm)

7	6 … 0
Reserved	FUNCADDR
0h	0-7Fh

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-201. Transmit Hub Port Register (USBn_TXHUBPORTm) Field Descriptions

Bit	Field	Description
7	Reserved	Reserved
6-0	HUBPORT	Port number of the hub

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.7.3.4 Receive Function Address Register (USBn_RXFUNCADDRm)

The receive function address register (USBn_RXFUNCADDRm) is a 7-bit read/write register (meaningful in host mode operation only) that record the address of the target function that is to be accessed through the associated endpoint (EPn). USBn_ RXFUNCADDR needs to be defined for each Rx endpoint that is used.

Note: USBn_RXFUNCADDRm need to be defined for Endpoint 0. This register (together with the USBn_RXHUBADDRm and USBn_RXHUBPORTm registers described below) allow the allocation of multiple devices to the USB controller endpoints.

The receive function address register is shown in Figure 15-185 and described in Table 15-202.

Figure 15-185. Receive Function Address Register (USBn_RXFUNCADDRm)

7	6 — 0
Reserved	FUNCADDR
0h	0-7Fh

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-202. Receive Function Address Register (USBn_RXFUNCADDRm) Field Descriptions

Bit	Field	Description
7	Reserved	Reserved
6-0	FUNCADDR	Address of target function.

15.5.7.3.5 Receive Hub Address Register (USBn_RXHUBADDRm)

The receive hub address register (USBn_RXHUBADDRm) is an 8-bit read/write register (meaningful in host mode operation only) which, like USBn_RXHUBPORTm, only needs to be written where a full- or low-speed device is connected to Rx Endpoint EPn via a high-speed USB 2.0 hub which carries out the necessary transaction translation to convert between high-speed transmission and full-/low-speed transmission. In such circumstances:

- The lower 7 bits should record the address of this USB 2.0 hub
- The top bit should record whether the hub has multiple transaction translators (set to ‘0’ if single transaction translator; set to ‘1’ if multiple transaction translators).

Note: If Endpoint 0 is connected to a hub, then USBn_RXHUBADDRm need to be defined for this endpoint.

The receive hub address register is shown in Figure 15-186 and described in Table 15-203.

Figure 15-186. Receive Hub Address Register (USBn_RXHUBADDRm)

7	6 — 0
Reserved	FUNCADDR
0h	0-7Fh

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-203. Receive Hub Address Register (USBn_RXHUBADDRm) Field Descriptions

Bit	Field	Description
7	MULT_TRANS	Set to 1 if hub has multiple transaction translators. Cleared to 0 if only single transaction translator is available.
6-0	HUBADDR	Address of HUB.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.7.3.6 Receive Hub Port Register (USBn_RXHUBPORTm)

The receive hub port register (USBn_RXHUBPORTm) (meaningful in host mode operation only) only needs to be written where a full- or low-speed device is connected to Rx Endpoint EPn via a high-speed USB 2.0 hub which carries out the necessary transaction translation. In such circumstances, these 7-bit read/write registers need to be used to record the port of that USB 2.0 hub through which the target associated with the endpoint is accessed.

Note: If Endpoint 0 is connected to a hub, then USBn_RXHUBPORT needs to be defined for this endpoint.

The receive hub port register is shown in Figure 15-187 and described in Table 15-204.

Figure 15-187. Receive Hub Port Register (USBn_RXHUBPORTm)

7	6 … 0
Reserved	FUNCADDR
0h	0-7Fh

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 15-204. Receive Hub Port Register (USBn_RXHUBPORTm) Field Descriptions

Bit	Field	Description
7	Reserved	Reserved
6-0	HUBPORT	Port number of HUB.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

15.5.7.4 Non-Indexed Endpoint Control/Status Registers

The registers available at indexed region, offset 10h–1Fh, are accessible independently of the setting of the Index register. The Non-Indexed region has dedicated block of register space for each endpoint with memory block between 100h–10Fh are reserved for endpoint 0 registers; memory block between 110h–11Fh are reserved for endpoint 1 registers; memory block between 120h–12Fh are reserved for endpoint 2 registers; and so on.

Table 15-205 and Table 15-206 display the set of endpoint registers that are accessible from their dedicated block of memory in both peripheral and host mode, respectively. Note that the definitions of these register is discussed in detail in Section 15.5.6.2.

Table 15-205. Peripheral Mode Configuration and Status Register Mapping where n=0,1 (USB0 or USB1) and m=1-15 (endpoint number)

Offset	Name	Description (within Indexed Region)
100h + 10h * m	USBn_TXMAXPm	See USBn_TXMAXP Reg Def
102h	USBn_ PERI_CSR0	See USBn_PERI_CSR0 Reg Def
102h + 10h * m	USBn_PERI_TXCSRm	See USBn_PERI_TXCSR Reg Def
104h + 10h * m	USBn_RXMAXPm	See USBn_RXMAXP Reg Def
106h + 10h * m	USBn_ PERI_RXCSRm	See USBn_PERI_RXCSR Reg Def
108h	USBn_COUNT0	See USBn_COUNT0 Reg Def
108h + 10h * m	USBn_ RXCOUNTm	See USBn_RXCOUNT Reg Def
10Fh	USBn_CONFIGDATA	See USBn_CONFIGDATA Reg Def

Table 15-206. Host Mode Configuration and Status Register Mapping where n=0,1 (USB0 or USB1) and m=1-15 (endpoint number)

Offset	Name	Description
100h +10h * m	USBn_TXMAXPm	See USBn_TXMAXP Reg Def
102h	USBn_HOST_CSR0	See USBn_HOST_CSR0 Reg Def
102h + 10h * m	USBn_HOST_TXCSRm	See USBn_HOST_TXCSR Reg Def
104h + 10h * m	USBn_RXMAXPm	See USBn_RXMAXP Reg Def
106h + 10h * m	USBn_HOST _RXCSRm	See USBn_HOST_RXCSR Reg Def
108h	USBn_COUNT0	See USBn_COUNT0 Reg Def
108h + 10h * m	USBn_RXCOUNTm	See USBn_RXCOUNT Reg Def
10Ah	USBn_HOST_TYPE0	See USBn_HOST_TYPE0 Reg Def
10Ah + 10h * m	USBn_HOST_TXTYPEm	See USBn_HOST_TXTYPE Reg Def
10Bh	USBn_HOST_NAKLIMIT0	See USBn_HOST_NAKLIMIT0 Reg Def
10Bh + 10h * m	USBn_HOST_TXINTERVALm	See USBn_HOST_TXINTERVAL Def
10Ch	USBn_HOST_TYPE0	See USBn_HOST_TYPE0 Reg Def
10Ch + 10h * m	USBn_HOST_RXTYPEm	See USBn_HOST_RXTYPE Reg Def
10Dh + 10h * m	USBn_HOST_RXINTERVALm	See USBn_HOST_RXINTERVAL Def
10Fh	USBn_CONFIGDATA	See USBn_CONFIGDATA Reg Def

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Chapter 16

SPRUH73–July 2011

Interprocessor Communication

This chapter describes the interprocessor communication of the device.

Topic **Page**

16.1 Mailbox **2280**

16.2 Spinlock **2282**

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.1 Mailbox

16.1.1 Introduction

16.1.1.1 Features

Global features of the Mailbox module are:

- OCP slave interface (L4) supports:
  - 32-bit data bus width
  - 8/16/32 bit access supported
  - 9-bit address bus width
  - Burst not supported
- 8 mailbox sub-modules
- Each mailbox sub module allows 1-way communication between 2 initiators
- Flexible mailbox/initiators assignment scheme
- 4 messages per mailbox sub-module
- 32-bit message width
- Support of 16/32-bit addressing scheme
- Non-intrusive emulation
- 4 interrupts (one per user: 1 to MPU Subsystem, 2 to PRUSS, and 1 to WakeM3)

16.1.1.2 Unsupported Features

There are no unsupported features for Mailbox on this device.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.1.2 Integration

This device contains a single instantiation of the Mailbox module at the system level. The mailbox function is made of eight sub-module mailboxes each supporting a 1-way communication between two initiators. The communication protocol from the sender to the receiver is implemented with mailbox registers using interrupts. The sender sends information to the receiver by writing to the mailbox. Interrupt signaling is used to notify the receiver a message has been queued or the sender for overflow situation.

The eight mailboxes are enough to handle communications between the MPU Subsystem, PRUSS PRUs, and WakeM3. Note that because the WakeM3 has access only to L4_Wakeup peripherals it does not have access to the Mailbox registers. A mailbox interrupt can still be sent to the M3 to trigger message notification. The actual message payload must be placed in either M3 internal memory or in the Control Module Interprocessor Message registers (IPC_MSG_REG{0-7}).

16.1.2.1 Mailbox Connectivity Attributes

The general connectivity for the Mailbox is shown in Table 16-1.

Table 16-1. Mailbox Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	L4PER_L4LS_GCLK
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	4 Interrupts mail_u0 – to MPU Subsystem mail_u1 – to PRUSS (PRU0) mail_u2 – to PRUSS (PRU1) mail_u3 – to WakeM3
DMA Requests	None
Physical Address	L4 Peripheral slave port

16.1.2.2 Mailbox Clock and Reset Management

The mailbox function operates from the L4 interface clock.

Table 16-2. Mailbox Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
Functional / Interface clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_l4ls_gclk From PRCM

16.1.2.3 Mailbox Pin List

The Mailbox module does not include any external interface pins.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2 Spinlock

16.2.1 Introduction

16.2.2 SPINLOCK Registers

Table 16-3 lists the memory-mapped registers for the SPINLOCK. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 16-3 should be considered as reserved locations and the register contents should not be modified.

Table 16-3. SPINLOCK REGISTERS

Offset	Acronym	Register Description	Section
0h	REV	Read-only IP revision identifier (X.Y.R) used by software to determine features, bugs and compatibility of an instance of this the Spin Lock module.	Section 16.2.2.1
10h	SYSCONFIG	This register controls the various parameters of the OCP interface. Note that several fields are present by read-only.	Section 16.2.2.2
14h	SYSTATUS	This register provides status information about this instance of the Spin Lock module.	Section 16.2.2.3
800h	LOCK_REG_0	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.4
804h	LOCK_REG_1	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.5
808h	LOCK_REG_2	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.6
80Ch	LOCK_REG_3	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.7
810h	LOCK_REG_4	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.8
814h	LOCK_REG_5	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.9
818h	LOCK_REG_6	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.10
81Ch	LOCK_REG_7	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.11

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 16-3. SPINLOCK REGISTERS (continued)

Offset	Acronym	Register Description	Section
820h	LOCK_REG_8	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.12
824h	LOCK_REG_9	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.13
828h	LOCK_REG_10	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.14
82Ch	LOCK_REG_11	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.15
830h	LOCK_REG_12	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.16
834h	LOCK_REG_13	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.17
838h	LOCK_REG_14	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.18
83Ch	LOCK_REG_15	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.19
840h	LOCK_REG_16	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.20
844h	LOCK_REG_17	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.21
848h	LOCK_REG_18	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.22
84Ch	LOCK_REG_19	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.23

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 16-3. SPINLOCK REGISTERS (continued)

Offset	Acronym	Register Description	Section
850h	LOCK_REG_20	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.24
854h	LOCK_REG_21	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.25
858h	LOCK_REG_22	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.26
85Ch	LOCK_REG_23	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.27
860h	LOCK_REG_24	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.28
864h	LOCK_REG_25	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.29
868h	LOCK_REG_26	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.30
86Ch	LOCK_REG_27	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.31
870h	LOCK_REG_28	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.32
874h	LOCK_REG_29	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.33
878h	LOCK_REG_30	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.34
87Ch	LOCK_REG_31	This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.	Section 16.2.2.35

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.1 REV Register (offset = 0h) [reset = 0h]

Read-only IP revision identifier (X.Y.R) used by software to determine features, bugs and compatibility of an instance of this the Spin Lock module.

REV is shown in Figure 16-1 and described in Table 16-4.

Figure 16-1. REV Register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REV																															
R																															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-4. REV Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	REV	R		IP Revision Code.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.2 SYSCONFIG Register (offset = 10h) [reset = 0h]

This register controls the various parameters of the OCP interface. Note that several fields are present by read-only.

SYSCONFIG is shown in Figure 16-2 and described in Table 16-5.

Figure 16-2. SYSCONFIG Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							CLOCKACTIVITY
R							R

7	6	5	4	3	2	1	0
Reserved			SIDLEMODE		ENWAKEUP	SOFTRESET	AUTOGATING
R			R		R	W	R

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-5. SYSCONFIG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-9	Reserved	R		
8	CLOCKACTIVITY	R		Indicates whether the module requires the OCP when in IDLE mode. 0 - NOTREQUIRED: OCP clock is not required by the module during IDLE mode and may be switched off. 1 - REQUIRED: OCP clock is required by the module, even during idle mode.
7-5	Reserved	R		
4-3	SIDLEMODE	R		Control of the slave interface power management IDLE request acknowledgement. 0 - FORCEIDLE: IDLE request is acknowledged unconditionally and immediately. 1 - NOIDLE: IDLE request is never acknowledged. 2 - SMARTIDLE: IDLE request acknowledgement is based on the internal module activity. 3 - unused3: Reserved. Do not use.
2	ENWAKEUP	R		Asynchronous wakeup gereration. 0 - NOWAKEUPGEN: Wakeup generation is disabled. 1 - WAKEUPGEN1: Enable wakeup generation.
1	SOFTRESET	W		Module software reset. 0 - NOOP: No Description 1 - DORESET: Start a soft reset sequence of the Spin Lock module.
0	AUTOGATING	R		Internal OCP clock gating strategy. 0 - FREERUNNING: OCP clock is not gated when OCP interface is idle. 1 - GATED: Automatic internal OCP clock gating strategy is applied, based on the OCP interface activity.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.3 SYSTATUS Register (offset = 14h) [reset = 0h]

This register provides status information about this instance of the Spin Lock module.

SYSTATUS is shown in Figure 16-3 and described in Table 16-6.

Figure 16-3. SYSTATUS Register

31	30	29	28	27	26	25	24
NUMLOCKS							
R							
23	22	21	20	19	18	17	16
Reserved							
R							
15	14	13	12	11	10	9	8
IU7	IU6	IU5	IU4	IU3	IU2	IU1	IU0
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
Reserved							RESETDONE
R							R

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-6. SYSTATUS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	NUMLOCKS	R		1 - 32LOCKS: This instance has 32 lock registers. 2 - 64LOCKS: This instance has 64 lock registers. 4 - 128LOCKS: This instance has 128 lock registers. 8 - 256LOCKS: This instance has 256 lock registers.
23-16	Reserved	R		
15	IU7	R		
14	IU6	R		
13	IU5	R		
12	IU4	R		
11	IU3	R		
10	IU2	R		
9	IU1	R		In-Use flag 1, covering lock registers 32 - 63. Reads as one only if one or more lock registers in this range are TAKEN. If no lock registers are implemented in this range, then this flag always reads as 0.
8	IU0	R		In-Use flag 0, covering lock registers 0 - 31. Reads as one only if one or more lock registers in this range are TAKEN.
7-1	Reserved	R		reserved
0	RESETDONE	R		0: Reset in progress. 1: Reset is completed.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.4 LOCK_REG_0 Register (offset = 800h) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_0 is shown in Figure 16-4 and described in Table 16-7.

Figure 16-4. LOCK_REG_0 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 16-7. LOCK_REG_0 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.5 LOCK_REG_1 Register (offset = 804h) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_1 is shown in Figure 16-5 and described in Table 16-8.

Figure 16-5. LOCK_REG_1 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-8. LOCK_REG_1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.6 LOCK_REG_2 Register (offset = 808h) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_2 is shown in Figure 16-6 and described in Table 16-9.

Figure 16-6. LOCK_REG_2 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-9. LOCK_REG_2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.7 LOCK_REG_3 Register (offset = 80Ch) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_3 is shown in Figure 16-7 and described in Table 16-10.

Figure 16-7. LOCK_REG_3 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-10. LOCK_REG_3 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.8 LOCK_REG_4 Register (offset = 810h) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_4 is shown in Figure 16-8 and described in Table 16-11.

Figure 16-8. LOCK_REG_4 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-11. LOCK_REG_4 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.9 LOCK_REG_5 Register (offset = 814h) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_5 is shown in Figure 16-9 and described in Table 16-12.

Figure 16-9. LOCK_REG_5 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-12. LOCK_REG_5 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.10 LOCK_REG_6 Register (offset = 818h) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_6 is shown in Figure 16-10 and described in Table 16-13.

Figure 16-10. LOCK_REG_6 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-13. LOCK_REG_6 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.11 LOCK_REG_7 Register (offset = 81Ch) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_7 is shown in Figure 16-11 and described in Table 16-14.

Figure 16-11. LOCK_REG_7 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-14. LOCK_REG_7 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.12 LOCK_REG_8 Register (offset = 820h) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_8 is shown in Figure 16-12 and described in Table 16-15.

Figure 16-12. LOCK_REG_8 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-15. LOCK_REG_8 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.13 LOCK_REG_9 Register (offset = 824h) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_9 is shown in Figure 16-13 and described in Table 16-16.

Figure 16-13. LOCK_REG_9 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-16. LOCK_REG_9 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.14 LOCK_REG_10 Register (offset = 828h) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_10 is shown in Figure 16-14 and described in Table 16-17.

Figure 16-14. LOCK_REG_10 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-17. LOCK_REG_10 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.15 LOCK_REG_11 Register (offset = 82Ch) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_11 is shown in Figure 16-15 and described in Table 16-18.

Figure 16-15. LOCK_REG_11 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-18. LOCK_REG_11 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.16 LOCK_REG_12 Register (offset = 830h) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_12 is shown in Figure 16-16 and described in Table 16-19.

Figure 16-16. LOCK_REG_12 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-19. LOCK_REG_12 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.17 LOCK_REG_13 Register (offset = 834h) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_13 is shown in Figure 16-17 and described in Table 16-20.

Figure 16-17. LOCK_REG_13 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-20. LOCK_REG_13 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.18 LOCK_REG_14 Register (offset = 838h) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_14 is shown in Figure 16-18 and described in Table 16-21.

Figure 16-18. LOCK_REG_14 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-21. LOCK_REG_14 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.19 LOCK_REG_15 Register (offset = 83Ch) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_15 is shown in Figure 16-19 and described in Table 16-22.

Figure 16-19. LOCK_REG_15 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-22. LOCK_REG_15 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.20 LOCK_REG_16 Register (offset = 840h) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_16 is shown in Figure 16-20 and described in Table 16-23.

Figure 16-20. LOCK_REG_16 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-23. LOCK_REG_16 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.21 LOCK_REG_17 Register (offset = 844h) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_17 is shown in Figure 16-21 and described in Table 16-24.

Figure 16-21. LOCK_REG_17 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-24. LOCK_REG_17 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.22 LOCK_REG_18 Register (offset = 848h) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_18 is shown in Figure 16-22 and described in Table 16-25.

Figure 16-22. LOCK_REG_18 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-25. LOCK_REG_18 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.23 LOCK_REG_19 Register (offset = 84Ch) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_19 is shown in Figure 16-23 and described in Table 16-26.

Figure 16-23. LOCK_REG_19 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-26. LOCK_REG_19 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.24 LOCK_REG_20 Register (offset = 850h) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_20 is shown in Figure 16-24 and described in Table 16-27.

Figure 16-24. LOCK_REG_20 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-27. LOCK_REG_20 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.25 LOCK_REG_21 Register (offset = 854h) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_21 is shown in Figure 16-25 and described in Table 16-28.

Figure 16-25. LOCK_REG_21 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-28. LOCK_REG_21 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.26 LOCK_REG_22 Register (offset = 858h) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_22 is shown in Figure 16-26 and described in Table 16-29.

Figure 16-26. LOCK_REG_22 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-29. LOCK_REG_22 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.27 LOCK_REG_23 Register (offset = 85Ch) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_23 is shown in Figure 16-27 and described in Table 16-30.

Figure 16-27. LOCK_REG_23 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-30. LOCK_REG_23 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.28 LOCK_REG_24 Register (offset = 860h) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_24 is shown in Figure 16-28 and described in Table 16-31.

Figure 16-28. LOCK_REG_24 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-31. LOCK_REG_24 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.29 LOCK_REG_25 Register (offset = 864h) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_25 is shown in Figure 16-29 and described in Table 16-32.

Figure 16-29. LOCK_REG_25 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-32. LOCK_REG_25 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.30 LOCK_REG_26 Register (offset = 868h) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_26 is shown in Figure 16-30 and described in Table 16-33.

Figure 16-30. LOCK_REG_26 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-33. LOCK_REG_26 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.31 LOCK_REG_27 Register (offset = 86Ch) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_27 is shown in Figure 16-31 and described in Table 16-34.

Figure 16-31. LOCK_REG_27 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-34. LOCK_REG_27 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.32 LOCK_REG_28 Register (offset = 870h) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_28 is shown in Figure 16-32 and described in Table 16-35.

Figure 16-32. LOCK_REG_28 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-35. LOCK_REG_28 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.33 LOCK_REG_29 Register (offset = 874h) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_29 is shown in Figure 16-33 and described in Table 16-36.

Figure 16-33. LOCK_REG_29 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-36. LOCK_REG_29 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.34 LOCK_REG_30 Register (offset = 878h) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_30 is shown in Figure 16-34 and described in Table 16-37.

Figure 16-34. LOCK_REG_30 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-37. LOCK_REG_30 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

16.2.2.35 LOCK_REG_31 Register (offset = 87Ch) [reset = 0h]

This register is read when attempting to acquire a lock. The lock is automatically taken if it was not taken and the value returned by the read is zero. If the lock was already taken, then the read returns one. Writing a zero to this register frees the lock.

LOCK_REG_31 is shown in Figure 16-35 and described in Table 16-38.

Figure 16-35. LOCK_REG_31 Register

31	30	29	28	27	26	25	24
Reserved							
R							

23	22	21	20	19	18	17	16
Reserved							
R							

15	14	13	12	11	10	9	8
Reserved							
R							

7	6	5	4	3	2	1	0
Reserved							TAKEN
R							R/W

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 16-38. LOCK_REG_31 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R		
0	TAKEN	R/W		0 - FREE: On a read, indicates that the lock was previously free. On a write, sets the lock to not taken (free). 1 - BUSY: On a read, indicates that the lock was already taken (busy). Writes of BUSY are ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Chapter 17

SPRUH73–July 2011

Multimedia Card (MMC)

This chapter describes the MMC of the device.

Topic **Page**

17.1 Introduction **2322**
17.2 Integration **2323**
17.3 Functional Description **2326**
17.4 Use Cases **2363**
17.5 MMC/SD/SDIO Registers **2364**

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.1 Introduction

17.1.1 MMCHS Features

The general features of the MMCHS host controller IP are:

- Built-in 1024-byte buffer for read or write
- Two DMA channels, one interrupt line
- Clock support
  - 96-MHz functional clock source input
  - up to 384Mbit/sec (48MByte/sec) in MMC mode 8-bit data transfer
  - up to 192Mbit/sec (24MByte/sec) in High-Speed SD mode 4-bit data transfer
  - up to 24Mbit/sec (3MByte/sec) in Default SD mode 1-bit data transfer
- Support for SDA 3.0 Part A2 programming model
- Serial link supports full compliance with:
  - MMC command/response sets as defined in the MMC standard specification v4.3.
  - SD command/response sets as defined in the SD Physical Layer specification v3.00
  - SDIO command/response sets and interrupt/read-wait suspend-resume operations as defined in the SD part E1 specification v 2 .00
  - SD Host Controller Standard Specification sets as defined in the SD card specification Part A2 v2.00

17.1.2 Unsupported MMCHS Features

The MMCHS module features not supported in this device are shown in Table 17-1.

Table 17-1. Unsupported MMCHS Features

Feature	Reason
MMC Out-of-band interrupts	MMC_OBI input tied low
Master DMA operation	Disabled through synthesis parameter
Card Supply Control (MMCHS(1-2))	Signal not pinned out

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.2 Integration

This device contains three instances of the Multimedia Card (MMC), Secure Digital (SD), and Secure Digital I/O (SDIO) high speed interface module (MMCHS). The controller provides an interface to an MMC, SD memory card or SDIO card.

The application interface is responsible for managing transaction semantics; the MMC/SDIO host controller deals with MMC/SDIO protocol at transmission level, packing data, adding CRC, start/end bit and checking for syntactical correctness. Figure 17-1 through Figure 17-3 below show examples of systems using the MMCHS controller. Note that the power switch control is only available on the MMCHS0 interface.

Figure 17-1. MMCHS Module SDIO Application

Figure 17-2. MMCHS SD (4-bit) Card Application

Figure 17-3. MMCHS Module MMC Application

17.2.1 MMCHS Connectivity Attributes

The general connectivity attributes for the three MMCHS modules are shown in Table 17-2.

Table 17-2. MMCHS Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_L4LS_GCLK (OCP) PD_PER_MMC_FCLK (Func) CLK_32KHZ (Debounce)
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	1 interrupt to Sabertooth
DMA Requests	2 DMA requests to EDMA (Active low, need to be inverted in glue logic)
Physical Address	L4 Peripheral slave port

17.2.2 MMCHS Clock and Reset Management

The MMCHS controller has separate bus interface and functional clocks. The debounce clock is created by dividing the 48-MHz (24 MHz @ OPP 50) clock in the PRCM by 2 and then dividing the resulting 24-MHz (12 MHz @ OPP50) clock by a fixed 732.4219 (366.2109 @ OPP 50) in the Control Module to get a 32-KHz clock. This clock is fed back into the PRCM for clock gating. (See the CLK32KDIVRATIO_CTRL register in Ch 5 for more details).

Table 17-3. MMCHS Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
CLK Interface clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_l4ls_gclk from PRCM
CLKADPI Functional clock	48 MHz	PER_CLKOUTM2 / 4	pd_per_mmc_fclk from PRCM
CLK32 Input de-bounce clock	32.768 KHz	CLK_24 / 732.4219	clk_32KHz from PRCM

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.2.3 MMCHS Pin List

The MMCHS interface pins are summarized in Table 17-4.

Table 17-4. MMCHS Pin List

Pin	Type	Description
MMC*x*_CLK	I/O	MMC/SD serial clock output
MMC*x*_CMD	I/O	MMC/SD command signal
MMC*x*_DAT0	I/O	MMC/SD data signal
MMC*x*_DAT1	I/O	MMC/SD data signal, SDIO interrupt input
MMC*x*_DAT2	I/O	MMC/SD data signal, SDIO read wait output
MMC*x*_DAT[7:3]	I/O	MMC/SD data signals
MMC*x*_POW	O	MMC/SD power supply control (MMCHS 0 only)
MMC*x*_SDCD	I	SD card detect (from connector)
MMC*x*_SDWP	I	SD write protect (from connector)
MMC*x*_OBI	I	MMC out of band interrupt

The direction of the data lines depends on the selected data transfer mode as summarized in Table 17-5.

Table 17-5. DAT Line Direction for Data Transfer Modes

	MMC/SD 1-bit mode	MMC/SD 4-bit mode	MMC/SD 8-bit mode	SDIO 1-bit mode	SDIO 4-bit mode
DAT[0]	I/O	I/O	I/O	I/O	I/O
DAT[1]	I[(1)]	I/O	I/O	I[(2)]	I/O or I[(2)]
DAT[2]	I[(1)]	I/O	I/O	I/O[(3)]	I/O or O[(3)]
DAT[3]	I[(1)]	I/O	I/O	I[(1)]	I/O
DAT[4]	I[(1)]	I[(1)]	I/O	I[(1)]	I[(1)]
DAT[5]	I[(1)]	I[(1)]	I/O	I[(1)]	I[(1)]
DAT[6]	I[(1)]	I[(1)]	I/O	I[(1)]	I[(1)]
DAT[7]	I[(1)]	I[(1)]	I/O	I[(1)]	I[(1)]

(1) Hi-Z state to avoid bus conflict.
(2) To support incoming interrupt from the SDIO card.
(3) To support read wait to the SDIO card. By default it is Input, Output only in read wait period.

The direction of the MMCHS data buffers are controlled by ADPDATDIROQ signals. ADPDATDIROQ[i] = 1 sets the corresponding DAT signal(s) in read position (input) and ADPDATDIROQ[i] = 0 sets the corresponding DAT signal(s) in write position (output). Additionally, the ADPDATDIRLS signals are provided (with opposite polarity) to control the direction of external level shifters. The value of these control signals for the various data modes are summarized in Table 17-6.

Table 17-6. ADPDATDIROQ and ADPDATDIRLS Signal States

	MMC/SD 1-bit mode	MMC/SD 4-bit mode	MMC/SD 8-bit mode	SDIO 1-bit mode	SDIO 4-bit mode
DAT[0]	ADPDATDIRLS[0] = 0 / 1 ADPDATDIROQ[0] = 1 / 0	ADPDATDIRLS[0] = 0 / 1 ADPDATDIROQ[0] = 1 / 0	ADPDATDIRLS[0] = 0 / 1 ADPDATDIROQ[0] = 1 / 0	ADPDATDIRLS[0] = 0 / 1	ADPDATDIRLS[0] = 0 / 1 ADPDATDIROQ[0] = 1 / 0
DAT[2]	ADPDATDIRLS[2] = 0 ADPDATDIROQ[2] = 1	ADPDATDIRLS[2] = 0 / 1 ADPDATDIROQ[2] = 1 / 0	ADPDATDIRLS[2] = 0 / 1 ADPDATDIROQ[2] = 1 / 0	ADPDATDIRLS[2] = 0 / 1 ADPDATDIROQ[2] = 1 / 0	ADPDATDIRLS[2] = 0 / 1 ADPDATDIROQ[2] = 1 / 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 17-6. ADPDATDIROQ and ADPDATDIRLS Signal States (continued)

	MMC/SD 1-bit mode	MMC/SD 4-bit mode	MMC/SD 8-bit mode	SDIO 1-bit mode	SDIO 4-bit mode
DAT[1]	ADPDATDIRLS[1] = 0 ADPDATDIROQ[1] = 1	ADPDATDIRLS[1] = 0 / 1 ADPDATDIROQ[1] = 1 / 0	ADPDATDIRLS[1] = 0 / 1 ADPDATDIROQ[1] = 1 / 0	ADPDATDIRLS[1] = 0 ADPDATDIROQ[1] = 1	ADPDATDIRLS[1] = 0 / 1 ADPDATDIROQ[1] = 1 / 0
DAT[3]					
DAT[4]	ADPDATDIRLS[3] = 0 ADPDATDIROQ[3] = 1	ADPDATDIRLS[3] = 0 ADPDATDIROQ[3] = 1	ADPDATDIRLS[3] = 0 / 1 ADPDATDIROQ[3] = 1 / 0	ADPDATDIRLS[3] = 0 ADPDATDIROQ[3] = 1	ADPDATDIRLS[3] = 0 ADPDATDIROQ[3] = 1
DAT[5]					
DAT[6]					
DAT[7]					

ADPDATIRLSx = 0 for input and 1 for output — these signals are not pinned out on this device.

ADPDATIROQx = 1 for output and 1 for input.

Grayed cells indicate that the data line is not used in the selected transfer mode.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.3 Functional Description

One MMC/SD/SDIO host controller can support one MMC memory card, one SD card, or one SDIO card.

Other combinations (for example, two SD cards, one MMC card, and one SD card) are not supported through a single controller.

17.3.1 MMC/SD/SDIO Functional Modes

17.3.1.1 MMC/SD/SDIO Connected to an MMC, an SD Card, or an SDIO Card

Figure 17-4 shows the MMC/SD/SDIO0 and MMC/SD/SDIO2 host controllers connected to an MMC, an SD, or an SDIO card and its related external connections.

Figure 17-4. MMC/SD 0/2 Connectivity to an MMC/SD Card

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

The following MMC/SD/SDIO controller pins are used

- **MMC_CMD** This pin is used for two-way communication between the connected card and the MMC/SD/SDIO controller. The MMC/SD/SDIO controller transmits commands to the card and the memory card drives responses to the commands on this pin.
- **MMC_DAT7-0** Depending on which type of card you are using, you may need to connect 1, 4, or 8 data lines. The number of DAT pins (the data bus width) is set by the Data Transfer Width (DTW) bit in the MMC control register (SD_HCTL). For more information, see the registers section of this document.
- **MMC_CLK** This pin provides the clock to the memory card from the MMC/SD controller.

  Note: The MMC_CLK pin functions as an output but must be configured as an I/O to internally loopback the clock to time the inputs.

Table 17-7 provides a summary of these pins.

Table 17-7. MMC/SD/SDIO Controller Pins and Descriptions

Pin	Type	1-Bit Mode	4-Bit Mode	8-Bit Mode	Reset Value
MMC_CLK[(1)]	O	Clock Line	Clock Line	Clock Line	High impedance
MMC_CMD	I/O	Command Line	Command Line	Command Line	High impedance
MMC_DAT0	I/O	Data Line 0	Data Line 0	Data Line 0	0
MMC_DAT1	I/O	(not used)	Data Line 1	Data Line 1	0
MMC_DAT2	I/O	(not used)	Data Line 2	Data Line 2	0
MMC_DAT3	I/O	(not used)	Data Line 3	Data Line 3	0
MMC_DAT4	I/O	(not used)	(not used)	Data Line 4	0
MMC_DAT5	I/O	(not used)	(not used)	Data Line 5	0
MMC_DAT6	I/O	(not used)	(not used)	Data Line 6	0
MMC_DAT7	I/O	(not used)	(not used)	Data Line 7	0

(1) The MMC_CLK pin functions as an output but must be configured as an I/O to internally loopback the clock to time the inputs.

Figure 17-5 shows the MMC/SD/SDIO1 host controller connected to an MMC, SD, or SDIO card and its related external connections. Note that MMC/SD/SDIO1 uses the same signals as MMC/SD/SDIO 0 and 2, but adds three additional signals. These are described below.

Figure 17-5. MMC/SD 1 Connectivity to an MMC/SD Card

Copyright © 2011, Texas Instruments Incorporated

The following MMC/SD/SDIO controller pins are used

- **MMC_POW** Used for MMC/SD card's cards on/off power supply control. When high, denotes power-on condition.
- **MMC_SDCD** This input pin serves as the MMC/SD/SDIO carrier detect. This signal is received from a mechanical switch on the slot.
- **MMC_SDWP** This input pin is used for the SD/SDIO card's write protect. This signal is received from a mechanical protect switch on the slot (system dependant). Applicable only for SD and SDIO cards that have a mechanical sliding tablet on the side of the card.

17.3.1.2 Protocol and Data Format

The bus protocol between the MMC/SD/SDIO host controller and the card is message-based. Each message is represented by one of the following parts:

Command: A command starts an operation. The command is transferred serially from the \MMC/SD/SDIO host controller to the card on the mmc_cmd line.

Response: A response is an answer to a command. The response is sent from the card to the MMC/SD/SDIO host controller. It is transferred serially on the mmc_cmd line.

Data: Data are transferred from the MMC/SD/SDIO host controller to the card or from a card to the MMC/SD/SDIO host controller using the DATA lines.

Busy: The mmc_dat0 signal is maintained low by the card as far as it is programming the data received.

CRC status: CRC result is sent by the card through the mmc_dat0 line when executing a write transfer. In the case of transmission error, occurring on any of the active data lines, the card sends a negative CRC status on mmc_dat0. In the case of successful transmission, over all active data lines, the card sends a positive CRC status on mmc_dat0 and starts the data programming procedure.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.3.1.2.1 Protocol

There are two types of data transfer:

- Sequential operation
- Block-oriented operation

There are specific commands for each type of operation (sequential or block-oriented). See the *Multimedia Card System Specification*, the *SD Memory Card Specifications*, and the *SDIO Card Specification, Part E1* for details about commands and programming sequences supported by the MMC, SD, and SDIO cards.

> **CAUTION**
>
> Stream commands are supported only by MMC cards.

Figure 17-6 and Figure 17-7 show how sequential operations are defined. Sequential operation is only for 1-bit transfer and initiates a continuous data stream. The transfer terminates when a stop command follows on the mmc_cmd line.

Figure 17-6. Sequential Read Operation (MMC Cards Only)

Figure 17-7. Sequential Write Operation (MMC Cards Only)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 17-8 and Figure 17-9 show how multiple block-oriented operations are defined. A multiple block-oriented operation sends a data block plus CRC bits. The transfer terminates when a stop command follows on the mmc_cmd line. These operations are available for all kinds of cards.

Figure 17-8. Multiple Block Read Operation (MMC Cards Only)

Figure 17-9. Multiple Block Write Operation (MMC Cards Only)

NOTE:

1. The card busy signal is not always generated by the card; the previous examples show a particular case.
2. It is the software's responsibility to do a software reset after a data timeout to ensure that mmc_clk is stopped. The software reset is done by setting bit 26 in the SD_SYSCTL register to 1.
3. For multiblock transfer, and especially for MMC cards, you can abort a transfer without using a stop command. Use a CMD23 before a data transfer to define the number of blocks that will be transferred, then the transfer stops automatically after the last block (provided the MMC card supports this feature).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.3.1.2.2 Data Format

Coding Scheme for Command Token

Command packets always start with 0 and end with 1. The second bit is a transmitter bit1 for a host command. The content is the command index (coded by 6 bits) and an argument (for example, an address), coded by 32 bits. The content is protected by 7-bit CRC checksum (see Figure 17-10).

Figure 17-10. Command Token Format

Coding Scheme for Response Token

Response packets always start with 0 and end with a 1. The second bit is a transmitter bit0 for a card response. The content is different for each type of response (R1, R2, R3, R4, R5, and R6) and the content is protected by 7-bit CRC checksum. Depending on the type of commands sent to the card, the SD_CMD register must be configured differently to avoid false CRC or index errors to be flagged on command response (see Table 17-8). For more details about response types, see the *Multimedia Card System Specification*, the *SD Memory Card Specification*, or the *SDIO Card Specification*.

Table 17-8. Response Type Summary[(1)]

Response Type SD_CMD[17:16] RSP_TYPE	Index Check Enable SD_CMD[20] CICE	CRC Check Enable SD_CMD[19] CCCE	Name of Response Type
00	0	0	No Response
01	0	1	R2
10	0	0	R3 (R4 for SD cards)
10	1	1	R1, R6, R5 (R7 for SD cards)
11	1	1	R1b, R5b

(1) The MMC/SD/SDIO host controller assumes that both clocks may be switched off, whatever the value set in the SD_SYSCONFIG[9:8] CLOCKACTIVITY bit.

Figure 17-11 and Figure 17-12 depict the 48-bit and 136-bit response packets.

Figure 17-11. 48-Bit Response Packet (R1, R3, R4, R5, R6)

Figure 17-12. 136-Bit Response Packet (R2)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Coding Scheme for Data Token

Data tokens always start with 0 and end with 1 (see Figure 17-13, Figure 17-14, Figure 17-15, and Figure 17-16).

Figure 17-13. Data Packet for Sequential Transfer (1-Bit)

Figure 17-14. Data Packet for Block Transfer (1-Bit)

Figure 17-15. Data Packet for Block Transfer (4-Bit)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 17-16. Data Packet for Block Transfer (8-Bit)

17.3.2 Resets

17.3.2.1 Hardware Reset

The module is reinitialized by the hardware.

The SD_SYSSTATUS[0] RESETDONE bit can be monitored by the software to check if the module is ready-to-use after a hardware reset.

This hardware reset signal has a global reset action on the module. All configuration registers and all state machines are reset in all clock domains.

This hardware reset signal has a global reset action on the module. All configuration registers and all state-machines are reset in all clock domains.

17.3.2.2 Software Reset

The module is reinitialized by software through the SD_SYSCONFIG[1] SOFTRESET bit. This bit has the same action on the module logic as the hardware signal except for:

- Debounce logic
- SD_PSTATE, SD_CAPA, and SD_CUR_CAPA registers (see corresponding register descriptions)

The SOFTRESET bit is active high. The bit is automatically reinitialized to 0 by the hardware. The SD_SYSCTL[24] SRA bit has the same action as the SOFTRESET bit on the design.

The SD_SYSSTATUS[0] RESETDONE bit can be monitored by the software to check if the module is ready-to-use after a software reset.

Copyright © 2011, Texas Instruments Incorporated

Moreover, two partial software reset bits are provided:

- SD_SYSCTL[26] SRD bit
- SD_SYSCTL[25] SRC bit

These two reset bits are useful to reinitialize data or command processes respectively in case of line conflict. When set to 1, a reset process is automatically released when the reset completes:

- The SD_SYSCTL[26] SRD bit resets all finite state-machines and status management that handle data transfers on both the interface and functional side.
- The SD_SYSCTL[25] SRC bit resets all finite state-machines and status management that handle command transfers on both the interface and functional side.

NOTE: If **any** of the clock inputs are not present for the MMC/SD/SDIO peripheral, the software reset will not complete.

17.3.3 Power Management

The MMC/SD/SDIO host controller can enter into different modes and save power:

- Normal mode
- Idle mode

The two modes are mutually exclusive (the module can be in normal mode or in idle mode). The MMC/SD/SDIO host controller is compliant with the PRCM module handshake protocol. When the MMC/SD/SDIO power domain is off, the only way to wake up the power domain and different MMC/SD/SDIO clocks is to monitor the mmc_dat1 input pin state via a different GPIO line for each MMC/SD/SDIO interface.

17.3.3.1 Normal Mode

The autogating of interface and functional clocks occurs when the following conditions are met:

- The SD_SYSCONFIG[0] AUTOIDLE bit is set to 1.
- There is no transaction on the MMC interface.

The autogating of interface and functional clocks stops when the following conditions are met:

- A register access occurs through the L3 (or L4) interconnect.
- A wake-up event occurs (an interrupt from a SDIO card).
- A transaction on the MMC/SD/SDIO interface starts.

Then the MMC/SD/SDIO host controller enters in low-power state even if SD_SYSCONFIG[0] AUTOIDLE is cleared to 0. The functional clock is internally switched off and only interconnect read and write accesses are allowed.

17.3.3.2 Idle Mode

The clocks provided to MMC/SD/SDIO are switched off upon a PRCM module request. They are switched back upon module request. The MMC/SD/SDIO host controller complies with the PRCM module handshaking protocol:

- Idle request from the system power manager
- Idle acknowledgment from the MMC/SD/SDIO host controller
- Wake-up request from the MMC/SD/SDIO host controller

The idle acknowledgment varies according to the SD_SYSCONFIG[4:3] SIDLEMODE bit field:

- 0: Force-idle mode. The MMC/SD/SDIO host controller acknowledges the system power manager request unconditionally.
- 1h: No-idle mode. The MMC/SD/SDIO host controller ignores the system power manager request and behaves normally as if the request was not asserted.
- 2h: Smart-idle mode. The MMC/SD/SDIO host controller acknowledges the system power manager

Copyright © 2011, Texas Instruments Incorporated

request according to its internal state.

- 3h: Smart-idle wake-up-capable mode. The MMC/SD/SDIO host controller acknowledges the system power manager request according to its internal state. However, the module may generate wake-up events when in idle state (related to IRQ or DMA requests)

During the smart-idle mode period, the MMC/SD/SDIO host controller acknowledges that the OCP and Functional clocks may be switched off whatever the value set in the SD_SYSCONFIG[9:8] CLOCKACTIVITY field.

17.3.3.3 Transition from Normal Mode to Smart-Idle Mode

Smart-idle mode is enabled when the SD_SYSCONFIG[4:3] SIDLEMODE bit field is set to 2h or 3h. The MMC/SD/SDIO host controller goes into idle mode when the PRCM issues an idle request, according to its internal activity. The MMC/SD/SDIO host controller acknowledges the idle request from the PRCM after ensuring the following:

- The current multi/single-block transfer is completed.
- Any interrupt or DMA request is asserted.
- There is no card interrupt on the SD_dat1 signal.

As long as the MMC/SD/SDIO controller does not acknowledge the idle request, if an event occurs, the MMC/SD/SDIO host controller can still generate an interrupt or a DMA request. In this case, the module ignores the idle request from the PRCM.

As soon as the MMC/SD/SDIO controller acknowledges the idle request from the PRCM:

- If Smart-Idle modethe module does not assert any new interrupt or DMA request
- If Smart-Idle wake-up-capable modethe module may generate wake-up events related to interrupt or DMA request.

17.3.3.4 Transition from Smart-Idle Mode to Normal Mode

The MMC/SD/SDIO host controller detects the end of the idle period when the PRCM deasserts the idle request. For the wake-up event, there is a corresponding interrupt status in the SD_STAT register. The MMC/SD/SDIO host controller operates the conversion between wake-up and interrupt (or DMA request) upon exit from smart-idle mode if the associated enable bit is set in the SD_ISE register.

Interrupts and wake-up events have independent enable/disable controls, accessible through the SD_HCTL and SD_ISE registers. The overall consistency must be ensured by software.

The interrupt status register SD_STAT is updated with the event that caused the wake-up in the CIRQ bit when the SD_IE[8] CIRQ_ENABLE associated bit is enabled. Then, the wake-up event at the origin of the transition from smart-idle mode to normal mode is converted into its corresponding interrupt or DMA request. (The SD_STAT register is updated and the status of the interrupt signal changes.)

When the idle request from the PRCM is deasserted, the module switches back to normal mode. The module is fully operational.

17.3.3.5 Force-Idle Mode

Force-idle mode is enabled when the SD_SYSCONFIG[4:3] SIDLEMODE bit field is cleared to 0. Force-idle mode is an idle mode where the MMC/SD/SDIO host controller responds unconditionally to the idle request from the PRCM. Moreover, in this mode, the MMC/SD/SDIO host controller unconditionally deasserts interrupts and DMA request lines are asserted.

The transition from normal mode to force-idle mode does not affect the bits of the SD_STAT register. In force-idle mode, the interrupt and DMA request lines are deasserted. Interface Clock (OCP) and functional clock (CLKADPI) can be switched off.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

CAUTION

In Force-ilde mode, an idle request from the PRCM during a command or a data transfer can lead to an unexpected and unpredictable result. When the module is idle, any access to the module generates an error as long as the OCP clock is alive.

The module exits the force-idle mode when the PRCM deassertes the idle request. Then the module switches back to normal mode. The module is fully operational. Interrupt and DMA request lines are optionally asserted one clock cycle later.

17.3.3.6 Local Power Management

Table 17-9 describes power-management features available for the MMC/SD/SDIO modules.

Table 17-9. Local Power Management Features

Feature	Registers	Description
Clock Auto Gating	SD_SYSCONFIG AUTOIDLE bit	This bit allows a local power optimization inside module, by gating the OCP clock upon the interface activity or gating the CLKADPI clock upon the internal activity.
Slave Idle Modes	SD_SYSCONFIG SIDLEMODE bit	Force-idle, No-idle, and Smart-idle modes
Clock Activity	SD_SYSCONFIG CLOCKACTIVITY bit	Please see Table 17-10 for configuration details.
Master Standby Modes	SD_SYSCONFIG STANDYBYMODE bit	Force-idle, No-idle, and Smart-idle modes
Global Wake-Up Enable	SD_SYSCONFIG ENAWAKEUP bit	This bit enables the wake-up feature at module level.
Wake-Up Sources Enable	SD_HCTL register	This register holds one active high enable bit per event source able to generate wake-up signal.

Table 17-10. Clock Activity Settings

CLOCKACTIVITY Values	Clock State When Module is in IDLE State		Features Available when Module is in IDLE State	Wake-Up Events
	OCP Clock	CLKADPI		
00	OFF	OFF	None	Card Interrupt
10	OFF	ON	None	
01	ON	OFF	None	
11	ON	ON	All	

CAUTION

The PRCM module has no hardware means of reading CLOCKACTIVITY settings. Thus, software must ensure consistent programming between the CLOCKACTIVITY and MMC clock PRCM control bits.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.3.4 Interrupt Requests

Several internal module events can generate an interrupt. Each interrupt has a status bit, an interrupt enable bit, and a signal status enable:

- The status of each type of interrupt is automatically updated in the SD_STAT register; it indicates which service is required.
- The interrupt status enable bits of the SD_IE register enable/disable the automatic update of the SD_STAT register on an event-by-event basis.
- The interrupt signal enable bits of the SD_ISE register enable/disable the transmission of an interrupt request on the interrupt line MMC_IRQ (from the MMC/SD/SDIOi host controller to the MPU subsystem interrupt controller) on an event-by-event basis.

If an interrupt status is disabled in the SD_IE register, then the corresponding interrupt request is not transmitted, and the value of the corresponding interrupt signal enable in the SD_ISE register is ignored.

When an interrupt event occurs, the corresponding status bit is automatically set to 1 (the MMC/SD/SDIO host controller updates the status bit) in the SD_STAT register. If later a mask is applied on the interrupt in the SD_ISE register, the interrupt request is deactivated.

When the interrupt source has not been serviced, if the interrupt status is cleared in the SD_STAT register and the corresponding mask is removed from the SD_ISE register, the interrupt status is not asserted again in the SD_STAT register and the MMC/SD/SDIOi host controller does not transmit an interrupt request.

CAUTION

If the buffer write ready interrupt (BWR) or the buffer read ready only interrupt (BRR) are not serviced and are cleared in the SD_STAT register, and the corresponding mask is removed, then the MMC/SD/SDIOi host controller will wait for the service of the interrupt without updating the status SD_STAT or transmitting an interrupt request.

Table 17-11 lists the event flags, and their mask, that can cause module interrupts.

Table 17-11. Events

Event Flag	Event Mask	Map To	Description
SD_STAT[29] BADA	SD_IE[29] BADA_ENABLE	MMC_IRQ	Bad Access to Data space. This bit is set automatically to indicate a bad access to buffer when not allowed. This bit is set during a read access to the data register (SD_DATA) while buffer reads are not allowed (SD_PSTATE[11] BRE=0). This bit is set during a write access to the data register (SD_DATA) while buffer writes are not allowed (SD_STATE[10] BWE=0)
SD_STAT[28] CERR	SD_IE[28] CERR_ENABLE	MMC_IRQ	Card Error. This bit is set automatically when there is at least one error in a response of type R1, R1b, R6, R5 or R5b. Only bits referenced as type E(error) in status field in the response can set a card status error. An error bit in the response is flagged only if corresponding bit in card status response errros SD_CSRE is set. There is not card detection for auto CMD12 command.
SD_STAT[25] ADMAE	SD_IE[25] ADMAE_ENABLE	MMC_IRQ	AMDA error. This bit is set when the host controller detects errors during ADMA based data transfer. The stat of the ADMA at an error occurrence is saved in the ADMA Error Status Register. In addition, the host controller generates this interrupt when it detects invalid descriptor data (Valid=0) at teh ST_FDS state.
SD_STAT[24] ACE	SD_IE[24] ACE_ENABLE	MMC_IRQ	Auto CMD12 error. This bit is set automatically when one of the bits in Auto CMD12 Error status register has changed from 0 to 1
SD_STAT[22] DEB	SD_IE[22] DEB_ENABLE	MMC_IRQ	Data End Bit error. This bit is set automatically when detecting a 0 at the end bit position of read data on DAT line or at the end position of the CRC status in write mode.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 17-11. Events (continued)

Event Flag	Event Mask	Map To	Description
SD_STAT[21] DCRC	SD_IE[21] DCRC_ENABLE	MMC_IRQ	Data CRC error. This bit is set automatically when there is a CRC16 error in the data phase response following a block read command or if there is a 3-bit CRC status different of a position "010" token during a block write command.
SD_STAT[20] DTO	SD_IE[20] DTO_ENABLE	MMC_IRQ	Data Timeout error. This bit is set automatically according to the following conidtions: A) busy timeout for R1b, R5b response. B) busty timeout after write CRC status. C) write CRC status timeout, or D) read data timeout.
SD_STAT[19] CIE	SD_IE[19] CIE_ENABLE	MMC_IRQ	Command Index Error. This bit is set automatically when response index differs from corresponding command index previously emitted. The check is enabled through SD_CMD[20] CICE bit.
SD_STAT[18] CEB	SD_IE[18] CEB_ENABLE	MMC_IRQ	Command End Bit error. This bit is set automatically when detecting a 0 at the end bit position of a command response.
SD_STAT[17] CCRC	SD_IE[17] CCRC_ENABLE	MMC_IRQ	Command CRC error. This bit is set automatically when there is a CRC7 error in the command response. CRC check is enabled through the SD_CMD[19] CCCE bit.
SD_STAT[16] CTO	SD_IE[16] CTO_ENABLE	MMC_IRQ	Command Timeout error. This bit is set automatically when no response is received within 64 clock cycles from the end bit of the command. For commands the reply within 5 clock cycles, the timeout is still detected at 64 clock cycles.
SD_STAT[15] ERRI	SD_IE[15] ERRI_ENABLE	MMC_IRQ	Error Interrupt. If any of the bits in the Error Interrupt Status register (SD_STAT[24:15]) are set, the this bit is set to 1.
SD_STAT[10] BSR	SD_IE[10] BSR_ENABLE	MMC_IRQ	Boot Status Receieved interrupt. This bit is set automatically when SD_CON[18] BOOT_CF0 is set to 1 or 2h and boot status is received on the dat0 line. This interrupt is only used for MMC cards.
SD_STAT[8] CIRQ	SD_IE[8] CIRQ_ENABLE	MMC_IRQ	Card Interrupt. This bit is only used for SD, SDIO, and CE-ATA cards. In 1-bit mode, interrupt source is asynchronous (can be a source of asynchronous wake-up). In 4-bit mode, interrupt source is sampled during the interrupt cycle. In CE-ATA mode, interrupt source is detected when the card drive CMD line to zero during one cycle after data transmission end.
SD_STAT[5] BRR	SD_IE[5] BRR_ENABLE	MMC_IRQ	Buffer Read ready. This bit is set automatically during a read operation to the card when one block specified by SD_BLK[10:0] BLEN is completely written in the buffer. It indicates that the memory card has filled out the buffger and the local host needs to empty the buffer by reading it.
SD_STAT[4] BWR	SD_IE[4] BWR_ENABLE	MMC_IRQ	Buffer Write ready. This bit is automatically set during a write operation to the card when the host can write a complete block as specified by SD_BLK[10:0] BLEN. It indicates that the memory card has emptied one block from the bugger and the local host is able to write one block of data into the buffer.
SD_STAT[3] DMA	SD_IE[3] DMA_ENABLE	MMC_IRQ	DMA interrupt. This status is set when an interrupt is required in the AMDA instruction and after the data transfer is complete.
SD_STAT[2] BGE	SD_IE[2] BGE_ENABLE	MMC_IRQ	Block Gap event. When a stop at block gap is requested (SD_HCTL[16] SBGR), this bit is automatically set when transaction is stopped at the block gap during a read or write operation.
SD_STAT[1] TC	SD_IE[1] TC_ENABLE	MMC_IRQ	Transfer completed. This bit is always set when a read/write transfer is completed or between two blocks when the transfer is stopped due to a stop at block gap requesd (SD_HCTL[16 SBGR). In read mode this bit is automatically set on completion of a read transfer (SD_PSTATE[9] RTA). In write mode, this bit is automatically set on completion of the DAT line use (SD_PSTATE[2] DLA).
SD_STAT[0] CC	SD_IE[0] CC_ENABLE	MMC_IRQ	Command complete. This bit is set when a 1-to-0 transition occurs in the register command inhibit (SD_PSTATE[0] CMDI). If the command is a type for which no response is expected, then the command complete interrupt is generated at the end of the command. A command timeout error (SD_STAT[16] CTO) has higher priority than command complete (SD_STAT[0] CC). If a response is expected but none is received, the a Command Timeout error is detected and signaled instead of the Command Complete interrupt.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.3.4.1 Interrupt-Driven Operation

An interrupt enable bit must be set in the SD_IE register to enable the module internal source of interrupt.

When an interrupt event occurs, the single interrupt line is asserted and the LH must:

- Read the SD_STAT register to identify which event occurred.
- Write 1 into the corresponding bit of the SD_STAT register to clear the interrupt status and release the interrupt line (if a read is done after this write, this would return 0).

NOTE: In the SD_STAT register, Card Interrupt (CIRQ) and Error Interrupt (ERRI) bits cannot be cleared.

The SD_STAT[8] CIRQ status bit must be masked by disabling the SD_IE[8] CIRQ_ENABLE bit (cleared to 0), then the interrupt routine must clear SDIO interrupt source in SDIO card common control register (CCCR).

The SD_STAT[15] ERRI bit is automatically cleared when all status bits in SD_STAT[31:16] are cleared.

17.3.4.2 Polling

When the interrupt capability of an event is disabled in the SD_ISE register, the interrupt line is not asserted:

- Software can poll the status bit in the SD_STAT register to detect when the corresponding event occurs.
- Writing 1 into the corresponding bit of the SD_STAT register clears the interrupt status and does not affect the interrupt line state.

NOTE: Please see the note in Section 17.3.4.1 concerning CIRQ and ERRI bits clearing.

17.3.5 DMA Modes

The device supports DMA slave mode only. In this case, the controller is slave on DMA transaction managed by two separated requests (SDMAWREQN and SDMARREQN)

17.3.5.1 DMA Slave Mode Operations

The MMC/SD/SDIO controller can be interfaced with a DMA controller. At system level, the advantage is to discharge the local host (LH) of the data transfers. The module does not support wide DMA access (above 1024 bytes) for SD cards as specified in the *SD Card Specification*and *SD Host Controller Standard Specification*.

The DMA request is issued if the following conditions are met:

- The SD_CMD[0] DE bit is set to 1 to trigger the initial DMA request (the write must be done when running the data transfer command).
- A command was emitted on the SD_cmd line.
- There is enough space in the buffer of the MMC/SD/SDIO controller to write an entire block (BLEN writes).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.3.5.1.1 DMA Receive Mode

In a DMA block read operation (single or multiple), the request signal SDMARREQN is asserted to its active level when a complete block is written in the buffer. The block size transfer is specified in the SD_BLK[10:0] BLEN field.

The SDMARREQN signal is deasserted to its inactive level when the sDMA has read one single word from the buffer. Only one request is sent per block; the DMA controller can make a 1-shot read access or several DMA bursts, in which case the DMA controller must manage the number of burst accesses, according to block size BLEN field.

New DMA requests are internally masked if the sDMA has not read exactly BLEN bytes and a new complete block is not ready. As DMA accesses are in 32-bit, then the number of sDMA read is Integer(BLEN/4)+1.

The receive buffer never overflows. In multiple block transfers for block size above 512 bytes, when the buffer gets full, the MMC_CLK clock signal (provided to the card) is momentarily stopped until the sDMA or the MPU performs a read access, which reads a complete block in the buffer.

Figure 17-17 provides a summary:

- DMA transfer size = BLEN buffer size in one shot or by burst
- One DMA request per block

Figure 17-17. DMA Receive Mode

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.3.5.1.2 DMA Transmit Mode

In a DMA block write operation (single or multiple), the request signal SDMAWREQN is asserted to its active level when a complete block is to be written to the buffer. The block size transfer is specified in the SD_BLK[10:0] BLEN field.

The SDMAWREQN signal is deasserted to its inactive level when the sDMA has written one single word to the buffer.

Only one request is sent per block; the DMA controller can make a 1-shot write access or multiple write DMA bursts, in which case the DMA controller must manage the number of burst accesses, according to block size BLEN field.

New DMA requests are internally masked if the sDMA has not written exactly BLEN bytes (as DMA accesses are in 32-bit, then the number of sDMA read is Integer(BLEN/4)+1) and if there is not enough memory space to write a complete block in the buffer.

Figure 17-18 provides a summary:

- DMA transfer size = BLEN buffer size in one shot or by burst
- One DMA request per block

Figure 17-18. DMA Transmit Mode

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.3.6 Mode Selection

The MMC/SD/SDIO host controller can be use in two modes, MMC and SD/SDIO modes. It has been designed to be the most transparent with the type of card. The type of the card connected is differentiated by the software initialization procedure.

Software identifies the type of card connected during software initialization. For each given card type, there are corresponding commands. Some commands are not supported by all cards. See the *Multimedia Card System Specification*, the *SD Memory Card Specifications*, and the *SDIO Card Specification, Part E1* for more details.

The purpose of the module is to transfer commands and data, to whatever card is connected, respecting the protocol of the connected card. Writes and reads to the card must respect the appropriate protocol of that card.

17.3.7 Buffer Management

17.3.7.1 Data Buffer

The MMC/SD/SDIO host controller uses a data buffer. This buffer transfers data from one data bus (Interconnect) to another data bus (SD, SDIO, or MMC card bus) and vice versa.

The buffer is the heart of the interface and ensures the transfer between the two interfaces (L4 and the card). To enhance performance, the data buffer is completed by a prefetch register and a post-write buffer that are not accessible by the host controller.

The read access time of the prefetch register is faster than the one of the data buffer. The prefetch register allows data to be read from the data buffer at an increased speed by preloading data into the prefetch register.

The entry point of the data buffer, the prefetch buffer, and the post-write buffer is the 32-bit register SD_DATA. A write access to the SD_DATA register followed by a read access from the SD_DATA register corresponds to a write access to the post-write buffer followed by a read access to the prefetch buffer. As a consequence, it is normal that the data of the write access to the SD_DATA register and the data of the read access to the SD_DATA register are different.

The number of 32-bit accesses to the SD_DATA register that are needed to read (or write) a data block with a size of SD_BLK[10:0] BLEN, and equals the rounded up result of BLEN divided by 4. The maximum block size supported by the host controller is hard-coded in the register SD_CAPA[17:16] MBL field and cannot be changed.

A read access to the SD_DATA register is allowed only when the buffer read enable status is set to 1 (SD_PSTATE[11] BRE); otherwise, a bad access (SD_STAT[29] BADA) is signaled.

A write access to the SD_DATA register is allowed only when the buffer write enable status is set to 1 (SD_PSTATE[10] BWE); otherwise, a bad access (SD_STAT[29] BADA) is signaled and the data is not written.

The data buffer has two modes of operation to store and read of the first and second portions of the data buffer:

- When the size of the data block to transfer is less than or equal to MEM_SIZE/2 (in double buffering), two data transfers can occur from one data bus to the other data bus and vice versa at the same time. The MMC/SD/SDIO controller uses the two portions of the data buffer in a ping-pong manner so that storing and reading of the first and second portions of the data buffer are automatically interchanged from time to time so that data may be read from one portion (for instance, through a DMA read access on the interconnect bus) while data (for instance, from the card) is being stored into the other portion and vice versa. When BLEN is less than or equal to 200h (that is, less or equal to 512Bytes), each of the two portions of the buffer that can be used have a size of BLEN (that is, 32-bits x BLEN div by 4). Not more than this total size of 2 times 32-bits × BLEN div by 4 can be used.
- When the size of the data block to transfer is larger than MEM_SIZE/2, only one data transfer can occur from one data bus to the other data bus at a time. The MMC/SD/SDIO host controller uses the entire data buffer as a single portion. In this mode, a bad access (SD_STAT[29] BADA) is signaled when two data transfers occur from one data bus to the other data bus and vice versa at the same time.

Copyright © 2011, Texas Instruments Incorporated

CAUTION

The SD_CMD[4] DDIR bit must be configured before a transfer to indicate the direction of the transfer.

Figure 17-19 shows the buffer management for writing and Figure 17-20 shows the buffer management for reading.

Figure 17-19. Buffer Management for a Write

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 17-20. Buffer Management for a Read

17.3.7.1.1 Memory Size, Block Length, and Buffer Management Relationship

The maximum block length and buffer management that can be targeted by system depend on memory depth setting.

Table 17-12. Memory Size, BLEN, and Buffer Relationship

Memory Size([5:2] MEMSIZE in bytes)	512	1024	2048	4096
Maximum block length supported	512	1024	2048	2048
Double-buffering for maximum block length	N/A	BLEN <= 512	BLEN <= 1024	BLEN <= 2048
Single-buffering for block length	BLEN<=512	512 < BLEN <= 1024	1024 < BLEN <= 2048	N/A

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.3.7.1.2 Data Buffer Status

The data buffer status is defined in the following interrupt status register and status register:

- Interrupt status registers (see Figure 17-58):
  - SD_STAT[29] BADABad access to data space
  - SD_STAT[5] BRRBuffer read ready
  - SD_STAT[4] BWRBuffer write ready
- Status registers (see Figure 17-55):
  - SD_PSTATE[11] BREBuffer read enable
  - SD_PSTATE[10] BWEBuffer write enable

17.3.8 Transfer Process

The process of a transfer is dependent on the type of command. It can be with or without a response, with or without data.

17.3.8.1 Different Types of Commands

Different types of commands are specific to MMC, SD, or SDIO cards. See the *Multimedia Card System Specification*, the *SD Memory Card Specifications*, the *SDIO Card Specification, Part E1*, or the *SD Card Specification, Part A2, SD Host Controller Standard Specification* for more details.

17.3.8.2 Different Types of Responses

Different types of responses are specific to MMC, SD, or SDIO cards. See the *Multimedia Card System Specification*, the *SD Memory Card Specifications*, the *SDIO Card Specification, Part E1*, or the *SD Card Specification, Part A2, SD Host Controller Standard Specification* for more details.

Table 17-13 shows how the MMC, SD, and SDIO responses are stored in the SD_RSPxx registers.

Table 17-13. MMC, SD, SDIO Responses in the SD_RSPxx Registers

Kind of Response	Response Field	Response Register
R1, R1b (normal response), R3, R4, R5, R5b, R6, R7	RESP[39:8][(1)]	SD_RSP10[31:0]
R1b (Auto CMD12 response)	RESP[39:8][(1)]	SD_RSP76[31:0]
R2	RESP[127:0][(1)]	SD_RSP76[31:0] SD_RSP54[31:0] SD_RSP32[31:0] SD_RSP10[31:0]

(1) RESP refers to the command response format described in the specifications mentioned above.

When the host controller modifies part of the SD_RSPxx registers, it preserves the unmodified bits.

The host controller stores the Auto CMD12 response in the SD_RSP76[31:0] register because the Host Controller may have a multiple block data DAT line transfer executing concurrently with a command. This allows the host controller to avoid overwriting the Auto CMD12 response with the command response stored in SD_RSP10 register and vice versa.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.3.9 Transfer or Command Status and Error Reporting

Flags in the MMC/SD/SDIO host controller show status of communication with the card:

- A timeout (of a command, a data, or a response)
- A CRC

Error conditions generate interrupts. See Table 17-14 and register description for more details.

Table 17-14. CC and TC Values Upon Error Detected

Error hold in the SD_STAT Register		CC	TC	Comments
29	BADA			No dependency with CC or TC. BADA is related to the register accesses. Its assertion is not dependent of the ongoing transfer.
28	CERR	1		CC is set upon CERR.
22	DEB		1	TC is set upon DEB.
21	DCRC		1	TC is set upon DCRC.
20	DTO			DTO and TC are mutually exclusive. DCRC and DEB cannot occur with DTO.
19	CIE	1		CC is set upon CIE.
18	CEB	1		CC is set upon CEB.
17	CCRC	1		CC can be set upon CCRC - See CTO comment
16	CTO			CTO and CC are mutually exclusive. CIE, CEB and CERR cannot occur with CTO. CTO can occur at the same time as CCRCit indicates a command abort due to a contention on CMD line. In this case no CC appears.

SD_STAT[21] DCRC event can be asserted in the following conditions:

- Busy timeout for R1b, R5b response type
- Busy timeout after write CRC status
- Write CRC status timeout
- Read data timeout
- Boot acknowledge timeout

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.3.9.1 Busy Timeout for R1b, R5b Response Type

Figure 17-21 shows DCRD event condition asserted when there is a busy timeout for R1b or R5b responses.

Figure 17-21. Busy Timeout for R1b, R5b Responses

t1 - Data timeout counter is loaded and starts after R1b, R5b response type.

t2 - Data timeout counter stops and if it is 0, SD_STAT[21] DCRC is generated.

17.3.9.2 Busy Timeout After Write CRC Status

Figure 17-22 shows DCRC event condition asserted when there is busy timeout after write CRC status.

Figure 17-22. Busy Timeout After Write CRC Status

t1 - Data timeout counter is loaded and starts after CRC status.

t2 - Data timeout counter stops and if it is 0, SD_STAT[21] DCRC is generated.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.3.9.3 Write CRC Status Timeout

Figure 17-23 shows DCRC event condition asserted when there is write CRC status timeout.

Figure 17-23. Write CRC Status Timeout

t1 - Data timeout counter is loaded and starts after Data block + CRC.

t2 - Data timeout counter stops and if it is 0, SD_STAT[21] DCRC is generated.

17.3.9.4 Read Data Timeout

Figure 17-24 shows DCRC event condition asserted when there is read data timeout.

Figure 17-24. Read Data Timeout

t1 - Data timeout counter is loaded and starts after Command transmission.

t2 - Data timeout counter stops and if it is 0, SD_STAT[21] DCRC is generated.

t3 - Data timeout counter is loaded and starts after Data block + CRC transmission.

t4 - Data timeout counter stops and if it is 0, SD_STAT[21] DCRC is generated.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.3.9.5 Boot Acknowledge Timeout

Figure 17-25 shows DCRC event condition asserted when there is boot acknowledge timeout and CMD0 is used.

Figure 17-25. Boot Acknowledge Timeout When Using CMD0

t1 - Data timeout counter is loaded and starts after CMD0.

t2 - Data timeout counter stops and if it is 0, SD_STAT[21] DCRC is generated.

t3 - Data timeout counter is loaded and starts.

t4 - Data timeout counter stops and if it is 0, SD_STAT[21] DCRC is generated.

t5 - Data timeout counter is loaded and starts after Data + CRC transmission.

t6 - Data timeout counter stops and if it is 0, SD_STAT[21] DCRC is generated.

Figure 17-26 shows DCRC event condition asserted when there is boot acknolwedge timeout and CMD line is held low.

Figure 17-26. Boot Acknowledge Timeout When CMD Held Low

t1 - Data timeout counter is loaded and starts after cmd line is tied to 0.

t2 - Data timeout counter stops and if it is 0, SD_STAT[21] DCRC is generated.

t3 - Data timeout counter is loaded and starts.

t4 - Data timeout counter stops and if it is 0, SD_STAT[21] DCRC is generated.

t5 - Data timeout counter is loaded and starts after Data + CRC transmission.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

t6 - Data timeout counter stops and if it is 0, SD_STAT[21] DCRC is generated.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.3.10 Auto Command 12 Timings

With the UHS definition of SD cards with higher frequency for MMC clocks up to 208, SD standard imposes a specific timing for Auto CMD12 "end bit" arrival.

17.3.10.1 Auto Command 12 Timings During Write Transfer

A margin named Ncrc in range of 2 to 8 cycles has been defined for SDR50 and SDR104 card components for write data transfers, as auto command 12 'end bit' shall arrive after the CRC status "end bit".

Figure 17-27 shows auto CMD12 timings during write transfer.

Figure 17-27. Auto CMD12 Timing During Write Transfer

The Host controller has a margin of 18 clock cycles to make sure that auto CMD12 'end bit' arrives after the CRC status. This margin does not depend on MMC bus configuration, DDR or standard transfer, 1,4 or 8 bus width.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.3.10.2 Auto Command 12 Timings During Read Transfer

With UHS very high speed cards gap timing between 2 successive cards has been extended to 4 cycles instead of 2. By the way it gives more flexibility for Host Auto CMD12 arrival in order to receive the last complete and reliable block. SD controller only follows the 'Left Border Case' defined by SD UHS specification.

Figure 17-28 shows ACMD12 timings during read transfer.

Figure 17-28. Auto Command 12 Timings During Read Transfer

The Auto CMD12 arrival sent by the Host controller is not sensitive to the MMC bus configuration whether it is DDR or standard transfer and whether it is a 1,4 or 8 bit bus width transfer.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.3.11 Transfer Stop

Whenever a transfer is initiated, the transmission may be willed to stop whereas it is still not finished. Several cases can be faced depending on the transfer type:

- Multiple blocks oriented transfers (for which transfer length is known)
- Continuous stream transfers (which have an infinite length)

NOTE: Since the MMC/SD/SDIO controller manages transfers based on a block granularity, the buffer will accept a block only if there is enough space to completely store it. Consequently, if a block is pending in the buffer, no command will be sent to the card because the card clock will be shut off by the controller.

The MMC/SD/SDIO controller includes two features which make a transfer stop more convenient and easier to manage:

- Auto CMD12 (for MMC and SD only).

  This feature is enabled by setting the SD_CMD[2] ACEN bit to 1 (this setting is relevant for a MMC/SD transfer with a known number of blocks to transfer). When the Auto CMD12 feature is enabled, the MMC/SD/SDIO controller will automatically issue a CMD12 command when the expected number of blocks has been exchanged.
- Stop at block gap

  This feature is enabled by setting the SD_HCTL[16] SBGR bit to 1. When enabled, this capability holds the transfer on until the end of a block boundary. If a stop transmission is needed, software can use this pause to send a CMD12 to the card.

Table 17-15 shows the common ways to stop a transfer, indicating command to send and features to enable.

Table 17-15. MMC/SD/SDIO Controller Transfer Stop Command Summary

		WRITE Transfer		READ Transfer	
		MMC/SD	SDIO	MMC/SD	SDIO
Single block		Transfer ends automatically Wait TC	Transfer ends automatically Wait TC	Transfer ends automatically Wait TC	Transfer ends automatically Wait TC
Multi blocks (finite or infinite)	Before the programmed block boundary	Send CMD12 Wait TC	Send CMD52 Wait TC	Send CMD12 Wait TC	Send CMD52 Wait TC
	Stop at the end of the transfer (finite transfer only)	Auto CMD12 active Transfer ends automatically Wait TC	Set SD_HCTL[16] SBGR bit to 1. Send CMD52 Wait TC	Auto CMD12 active Transfer ends automatically Wait TC	**If READ_WAIT supported** Stop at block gap Wait TC
					If READ_WAIT not supported Send CMD52 Wait TC

NOTE: The MMC/SD/SDIO controller will send the stop command to the card on a block boundary, regardless the moment the command was written to the controller registers.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.3.12 Output Signals Generation

The MMC/SD/SDIO output signals can be driven on either falling edge or rising edge depending on the SD_HCTL[2] HSPE bit. This feature allows to reach better timing performance, and thus to increase data transfer frequency.

17.3.12.1 Generation on Falling Edge of MMC Clock

The controller is by default in this mode to maximize hold timings. In this case, SD_HCTL[2] HSPE bit is cleared to 0.

Figure 17-29 shows the output signals of the module when generating from the falling edge of the MMC clock.

Figure 17-29. Output Driven on Falling Edge

17.3.12.2 Generation on Rising Edge of MMC Clock

This mode increases setup timings and allows reaching higher bus frequency. This feature is activated by setting SD_HCTL[2] HSPE bit to 1. The controller shall be set in this mode to support SDR transfers.

NOTE: Do not use this feature in Dual Data Rate mode (when SD_CON[19] DDR is set to 1).

Figure 17-30 shows the output signals of the module when generating from the rising edge of the MMC clock.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 17-30. Output Driven on Rising Edge

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.3.13 Card Boot Mode Management

Boot Operation Mode allows the MMC/SD/SDIO host controller to read boot data from the connected slave (MMC device) by keeping CMD line low after power-on (or sending CMD0 with specific argument) before issuing CMD1. The data can be read from either boot area or user area, depending on register setting. Power-on boot defines a way for the boot-code to be accessed by the MMC/SD/SDIO host controller without an upper-level software driver, speeding the time it takes for a controller to access the boot code.

The two possible ways to issue a boot command (either issuing a CMD0 or driving the CMD line to 0 during the whole boot phase) are described in the following sections.

17.3.13.1 Boot Mode Using CMD0

Figure 17-31 shows the timing diagram of a boot sequence using CMD0.

Figure 17-31. Boot Mode With CMD0

* Refer to MMC Specification for correct argument.

17.3.13.2 Boot Mode With CMD Held Low

Figure 17-32 shows the timing diagram of a boot sequence with CMD line tied to 0.

Figure 17-32. Boot Mode With CMD Line Tied to 0

For more information on how to configure the MMC/SD/SDIO host controller, see the section titled Boot with CMD line tied to 0.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.3.14 CE-ATA Command Completion Disable Management

The MMC/SD/SDIO controller supports CE-ATA features, in particular the detection of command completion token. When a command that requires a command completion signal (SD_CON[12] CEATA and SD_CMD[2] ACEN set to 1) is launched, the host system is no longer allowed to emit a new command in parallel of data transfer unless it is a command completion disable token.

The settings to emit a command completion disable token follow:

- SD_CON[12] CEATA is set to 1.
- SD_CON[2] HR set to 1.
- Clear the SD_ARG register.
- Write into SD_CMD register with value 0000 0000h.

When a command completion disable token was emitted (that is, SD_STAT[0] CC received), the host system is again allowed to emit another type of command (for example a transfer abort command CMD12 to abort transfer).

A critical case can be met when command completion signal disable (CCSD) is emitted during the last data block transfer, the sequence on command line could be sent very close to command completion signal (CCS) token sent by the card.

Three cases can be met:

- CCS is receive just before CCSD is emitted:

  An interrupt CIRQ is generated with CCS detection, CCSD is transmitted to card then an interrupt CC is generated when CCSD ends. In this case, card consider the CCSD sequence.
- CCS is not generated or generated during the CCSD transfer:

  The CCS bit cannot be detected (conflict is not possible as they drive the same level on command line, then no CIRQ interrupt is generated; besides CC interrupt is generated when CCSD ends).
- CCS is generated without CCSD token required:

  Only the interrupt CIRQ is generated when CCS is detected.

17.3.15 Test Registers

Test registers are available to be compliant with SD Host controller specification. This feature is useful to generate interrupts manually for driver debugging. The Force Event register (SD_FE) is used to control the Error Interrupt Status and Auto CMD12 Error Status. The System Test register (SD_SYSTEST) is used to control the signals that connect to I/O pins when the module is configured in system test (SD_CON[4] MODE = 1) mode for boundary connectivity verification.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.3.16 MMC/SD/SDIO Hardware Status Features

Table 17-16 summarizes the MMC/SD/SDIO hardware status features.

Table 17-16. MMC/SD/SDIO Hardware Status Features

Feature	Type	Register/Bit Field/Observability Control	Description
Interrupt flags		See Section 17.3.4.	
CMD line signal level	Status	[24] CLEV	Indicates the level of the cmd line
DAT lines signal level	Status	[23:20] DLEV	Indicates the level of the data lines
Buffer read enable	Status	[11] BRE	Readable data exists in the buffer.
Buffer write enable	Status	[10] BWE	Indicates whether there is enough space in the buffer to write BLEN bytes of data
Read transfer active	Status	[9] RTA	This status is used for detecting completion of a read transfer.
Write transfer active	Status	[8] WTA	This status indicates a write transfer active.
Data line active	Status	[2] DLA	Indicates whether the data lines are active
Command Inhibit (data lines)	Status	[1] DATI	Indicates whether issuing of command using data lines is allowed
Command inhibit (CMD line)	Status	[0] CMDI	Indicates whether issuing of command using CMD line is allowed

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.3.17 Low-Level Programming Models

17.3.17.1 Surrounding Modules Global Initialization

This section identifies the requirements of initializing the surrounding modules when the module has to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the MMC/SD/SDIO modules.

Table 17-17. Global Init for Surrounding Modules

Surrounding Modules	Comments
PRCM	Module interface and functional clocks must be enabled. For more information, see Power, Reset, and Clock Management (PRCM) chapter
Control module	Module-specific pad muxing and configuration must be set in the control module. See Control Module chapter.
(optional) MPU INTC	MPU INTC configuration must be done to enable the interrupts from the SD module. See Interrupt Controller chapter
(optional) sDMA (or dDMA)	DMA configuration must be done to enable the module DMA channel requests. See SDMA chapter
(optional) Interconnect	For more information about the interconnect configuration, see Interconnect chapter

NOTE: The MPU interrupt controller and the sDMA/dDMA configurations are necessary, if the interrupt and DMA based communication modes are used.

17.3.17.2 MMC/SD/SDIO Controller Initialization Flow

The next sections outline the four steps to initialize the MMC/SD/SDIO controller:

- Initialize Clocks
- Software reset of the controller
- Set module's hardware capabilities
- Set module's Idle and Wake-Up modes

17.3.17.2.1 Enable OCP and CLKADPI Clocks

Prior to any SD register access one must enable the SD OCP clock and CLKADPI clock in PRCM module registers. Refer to the *Power, Reset, and Clock Management* chapter.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.3.17.2.2 SD Soft Reset Flow

Figure 17-33 shows the soft reset process of MMC/SD/SDIO controller.

Figure 17-33. MMC/SD/SDIO Controller Software Reset Flow

17.3.17.2.3 Set SD Default Capabilities

Software must read capabilities (in boot ROM for instance) and is allowed to set (write) SD_CAPA[26:24] and SD_CUR_CAPA[23:0] registers before the MMC/SD/SDIO host driver is started.

17.3.17.2.4 Wake-Up Configuration

Table 17-18 details SD controller wake-up configuration.

Table 17-18. MMC/SD/SDIO Controller Wake-Up Configuration

Step	Access Type	Register/Bit Field/Programming Model
Configure wake-up bit (if necessary).	W	SD_SYSCONFIG[2] ENAWAKEUP
Enable wake-up events on SD card interrupt (if necessary).	W	SD_HCTL[24] IWE
SDIO Card onlyEnable card interrupt (if necessary).	W	SD_IE[8] CIRQENABLE

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.3.17.2.5 MMC Host and Bus Configuration

Figure 17-34 details the MMC bus configuration process.

Figure 17-34. MMC/SD/SDIO Controller Bus Configuration Flow

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.3.17.3 Operational Modes Configuration

17.3.17.3.1 Basic Operations for MMC/SD/SDIO Host Controller

The MMC/SD/SDIO controller performs data transfers: data to card (referred to as write transfers) and data from card (referred to as read transfers).

The host controller requires transfers to run on a block-by-block basis, rather than on a DMA burst size basis. A single DMA request (or block request interrupt) is signaled for each block. Pipelining is supported as long as the block size is less than one half of the memory buffer size.

17.3.17.3.2 Card Detection, Identification, and Selection

Figure 17-35 and Figure 17-36 show the card identification and selection process.

Figure 17-35. MMC/SD/SDIO Controller Card Identification and Selection - Part 1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 17-36. MMC/SD/SDIO Controller Card Identification and Selection - Part 2

17.4 Use Cases

Copyright © 2011, Texas Instruments Incorporated

17.5 MMC/SD/SDIO Registers

Table 17-19 lists the MMC/SD/SDIO registers.

CAUTION

The MMC/SD/SDIO registers are limited to 32-bit data accesses. 16-bit and 8-bit are not allowed and can corrupt register content.

Table 17-19. MMC/SD/SDIO Registers

Address Offset	Acronym	Register Name	Section
0	SD_HL_REV	IP Revision Identifier	Section 17.5.1
4h	SD_HL_HWINFO	Hardware Configuration	Section 17.5.2
10h	SD_HL_SYSCONFIG	Clock Management Configuration	Section 17.5.3
110h	SD_SYSCONFIG	System Configuration	Section 17.5.4
114h	SD_SYSSTATUS	System Status	Section 17.5.5
124h	SD_CSRE	Card status response error	Section 17.5.6
128h	SD_SYSTEST	System Test	Section 17.5.7
12Ch	SD_CON	Configuration	Section 17.5.8
130h	SD_PWCNT	Power counter	Section 17.5.9
200h	SD_SDMASA	SDMA System address:	Section 17.5.10
204h	SD_BLK	Transfer Length Configuration	Section 17.5.11
208h	SD_ARG	Command argument	Section 17.5.12
20Ch	SD_CMD	Command and transfer mode	Section 17.5.13
210h	SD_RSP10	Command Response 0 and 1	Section 17.5.14
214h	SD_RSP32	Command Response 2 and 3	Section 17.5.15
218h	SD_RSP54	Command Response 4 and 5	Section 17.5.16
21Ch	SD_RSP76	Command Response 6 and 7	Section 17.5.17
220h	SD_DATA	Data	Section 17.5.18
224h	SD_PSTATE	Present state	Section 17.5.19
228h	SD_HCTL	Host Control	Section 17.5.20
22Ch	SD_SYSCTL	SD system control	Section 17.5.21
230h	SD_STAT	SD interrupt status	Section 17.5.22
234h	SD_IE	SD interrupt enable	Section 17.5.23
238h	SD_ISE	SD interrupt enable set	Section 17.5.24
23Ch	SD_AC12	Auto CMD12 Error Status	Section 17.5.25
240h	SD_CAPA	Capabilities	Section 17.5.26
248h	SD_CUR_CAPA	Maximum current capabilities	Section 17.5.27
250h	SD_FE	Force Event	Section 17.5.28
254h	SD_ADMAES	ADMA Error Status	Section 17.5.29
258h	SD_ADMASAL	ADMA System address Low bits	Section 17.5.30
25Ch	SD_ADMASAH	ADMA System address High bits	Section 17.5.31
2FCh	SD_REV	Versions	Section 17.5.32

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.5.1 IP Revision Identifier Register (SD_HL_REV)

This register holds the IP Revision Identifier (X.Y.R) information.

Figure 17-37. SD_HL_REV Register

31–30	29–28	27–16
SCHEME	Reserved	FUNC
R-0	R-0	R-20h

15–10	8–7	6–5	4–0
R_RTL	X_MAJOR	CUSTOM	Y_MINOR
R-0	R-x	R-0	R-x

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

17.5.2 IP Module Hardware Configuration Register (SD_HL_HWINFO)

This register provides information about the IP module's hardware configuration, i.e. typically the module's HDL generics (if any).

Figure 17-38. SD_HL_HWINFO Register

31–16
Reserved
R-0

15–2	1	0
Reserved	MEM_SIZE	MADMA_EN
R-0	R-0	R-x

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

17.5.3 Clock Management Configuration (SD_HL_SYSCONFIG)

This register provides information about the clock management configuration.

Figure 17-39. SD_HL_SYSCONFIG Register

31–16
Reserved
R-0

15–6	5–4	3–2	1	0
Reserved	STANDBYMODE	IDLEMODE	FREEEMU	SOFTRESET
R-0	R/W-2h	R/W-2h	R/W-0	R/W-0

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.5.4 System Configuration Register (SD_SYSCONFIG)

This register allows controlling various parameters of the OCP interface. The system configuration register (SD_SYSCONFIG) is shown in Figure 17-40 and described in Table 17-20.

Figure 17-40. System Configuration Register (SD_SYSCONFIG)

31							16
Reserved							
R-0							
15	14	13	12	11	10	9	8
Reserved		STANDBYMODE		Reserved		CLOCKACTIVITY	
R-0		R/W-2h		R-0		R/W-0	

7	5	4	3	2	1	0
Reserved		SIDLEMODE		ENAWAKEUP	SOFTRESET	AUTOIDLE
R-0		R/W-2h		R/W-1	R/W-0	R/W-1

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

Table 17-20. System Configuration Register (SD_SYSCONFIG) Field Descriptions

Bit	Field	Value	Description
31-14	Reserved	0	These bits are initialized to zero, and writes to them are ignored. Reads return 0.
13-12	STANDBYMODE		Master interface power Management, standby/wait control. The bit field is only useful when generic parameter MADMA_EN (Master ADMA enable) is set as active, otherwise it is a read only register read a 0.
		0	Force-standby. Mstandby is forced unconditionally.
		1h	No-standby. Mstandby is never asserted.
		2h	Smart-standby modelocal initiator standby status depends on local conditions, i.e. the module's functional requirement from the initiator. IP module shall not generate (initiator-related) wake-up events.
		3h	Smart-Standby wake-up-capable modelocal initiator standby status depends on local conditions, i.e. the module's functional requirement from the initiator. IP module can generate (master-related) wake-up events when in standby state. Mode is only relevant if the appropriate IP module "mwake-up" output is implemented. Functional clock is maintained. Interface clock may be switched off.
11-10	Reserved	0	These bits are initialized to zero, and writes to them are ignored. Reads return 0.
9-8	CLOCKACTIVITY		Clocks activity during wake up mode period.
		Bit 8	Interface clock
		Bit 9	Functional clock
		0	Interface and Functional clock may be switched off.
		1h	Interface clock is maintained. Functional clock may be switched-off.
		2h	Functional clock is maintained. Interface clock may be switched-off.
		3h	Interface and Functional clocks are maintained.
7-5	Reserved	0	These bits are initialized to zero, and writes to them are ignored. Reads return 0.
4-3	SIDLEMODE		Power management
		0	If an idle request is detected, the MMC/SD/SDIO host controller acknowledges it unconditionally and goes in Inactive mode. Interrupt and DMA requests are unconditionally deasserted.
		1h	If an idle request is detected, the request is ignored and the module keeps on behaving normally.
		2h	If an idle request is detected, the module will switch to wake up mode based on its internal activity, and the wake up capability can be used if the wake up capability is enabled (bit SD_SYSCONFIG[2] ENAWAKEUP bit is set to 1).
		3h	Smart-idle Wake-up capable modelocal target's idle state eventually acknowledges the system's idle requests, depending on the IP modules internal requirements. IP module may generate (IRQ- or DMA-request-related) wake-up events when in idle state. Mode is only relevant if the appropriate IP module "swake-up" output(s) is (are) implemented.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 17-20. System Configuration Register (SD_SYSCONFIG) Field Descriptions (continued)

Bit	Field	Value	Description
2	ENAWAKEUP		Wake-up feature control
		0	Wake-up capability is disabled.
		1	Wake-up capability is enabled.
1	SOFTRESET		Software reset. The bit is automatically reset by the hardware. During reset, it always returns 0.
		Read 0	Normal mode
		Write 0	No effect
		Read 1	The module is reset.
		Write 1	Trigger a module reset.
0	AUTOIDLE		Internal Clock gating strategy
		Read 0	Clocks are free-running.
		Write 1	Automatic clock gating strategy is applied, based on the interconnect and MMC interface activity.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.5.5 System Status Register (SD_SYSSTATUS)

This register provides status information about the module excluding the interrupt status information. The system status register (SD_SYSSTATUS) is shown in Figure 17-41 and described in Table 17-21.

Figure 17-41. System Status Register (SD_SYSSTATUS)

31	1	0
Reserved		RESETDONE
R-0		R-0

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

Table 17-21. System Status Register (SD_SYSSTATUS) Field Descriptions

Bit	Field	Value	Description
31-1	Reserved	0	Reserved bit field. Do not write any value.
0	RESETDONE		Internal Reset Monitoring. Notethe debounce clock , the interface clock and the functional clock shall be provided to the MMC/SD/SDIO host controller to allow the internal reset monitoring.
		Read 0	Internal module reset is on-going
		Read 1	Reset completed

17.5.6 Card Status Response Error (SD_CSRE)

This register enables the host controller to detect card status errors of response type R1, R1b for all cards and of R5, R5b and R6 response for cards types SD or SDIO. When a bit SD_CSRE[i] is set to 1, if the corresponding bit at the same position in the response SD_RSP10[I] is set to 1, the host controller indicates a card error (SD_STAT[28] CERR bit) interrupt status to avoid the host driver reading the response register (SD_RSP10).

NOTE: No automatic card error detection for autoCMD12 is implemented; the host system has to check autoCMD12 response register (SD_RSP76) for possible card errors.

The card status response error (SD_CSRE) is shown in Figure 17-42 and described in Table 17-22.

Figure 17-42. Card Status Response Error (SD_CSRE)

31	0
CSRE	
R/W-0	

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

Table 17-22. Card Status Response Error (SD_CSRE) Field Descriptions

Bit	Field	Value	Description
31-0	CSRE		Card status response error

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.5.7 System Test Register (SD_SYSTEST)

This register is used to control the signals that connect to I/O pins when the module is configured in system test (SYSTEST) mode for boundary connectivity verification.

NOTE: In SYSTEST mode, a write into SD_CMD register will not start a transfer. The buffer behaves as a stack accessible only by the local host (push and pop operations). In this mode, the Transfer Block Size (SD_BLK[10:0] BLEN bits) and the Blocks count for current transfer (SD_BLK[31:16] NBLK bits) are needed to generate a Buffer write ready interrupt (SD_STAT[4] BWR bit) or a Buffer read ready interrupt (SD_STAT[5] BRR bit) and DMA requests if enabled.

The system test register (SD_SYSTEST) is shown in Figure 17-43 and described in Table 17-23.

Figure 17-43. System Test Register (SD_SYSTEST)

31							17	16
Reserved								OBI
			R-0					R/W-0

15	14	13	12	11	10	9	8
SDCD	SDWP	WAKD	SSB	D7D	D6D	D5D	D4D
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

7	6	5	4	3	2	1	0
D3D	D2D	D1D	D0D	DDIR	CDAT	CDIR	MCKD
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

Table 17-23. System Test Register (SD_SYSTEST) Field Descriptions

Bit	Field	Value	Description
31-17	Reserved	0	Reserved bit field. Do not write any value. Reads return 0.
16	OBI		Out-of-band interrupt (OBI) data value.
		0	The out-of-band interrupt pin is driven low.
		1	The out-of-band interrupt pin is driven high.
15	SDCD		Card detect input signal (SDCD) data value
		0	The card detect pin is driven low.
		1	The card detect pin is driven high.
14	SDWP		Write protect input signal (SDWP) data value
		0	The write protect pin SDWP is driven low.
		1	he write protect pin SDWP is driven high.
13	WAKD		Wake request output signal data value
		Read 0	No action. Returns 0.
		Write 0	The pin SWAKEUP is driven low.
		Read 1	No action. Returns 1.
		Write 1	The pin SWAKEUP is driven high.
12	SSB		Set status bit. This bit must be cleared prior attempting to clear a status bit of the interrupt status register (SD_STAT).
		Read 0	No action. Returns 0.
		Write 0	Clear this SSB bit field. Writing 0 does not clear already set status bits.
		Read 1	No action. Returns 1.
		Write 1	Force to 1 all status bits of the interrupt status register (SD_STAT) only if the corresponding bit field in the Interrupt signal enable register (SD_ISE) is set.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 17-23. System Test Register (SD_SYSTEST) Field Descriptions (continued)

Bit	Field	Value	Description
11	D7D		DAT7 input/output signal data value
		Read 0	If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT7 line (low). If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 0.
		Write 0	If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT7 line is driven low. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect.
		Read 1	If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT7 line (high) If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 1.
		Write 1	If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT7 line is driven high. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect.
10	D6D		DAT6 input/output signal data value
		Read 0	If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT6 line (low). If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 0.
		Write 0	If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT6 line is driven low. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect.
		Read 1	If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT6 line (high) If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 1.
		Write 1	If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT6 line is driven high. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect.
9	D5D		DAT5 input/output signal data value
		Read 0	If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT5 line (low). If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 0.
		Write 0	If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT5 line is driven low. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect.
		Read 1	If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT5 line (high) If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 1.
		Write 1	If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT5 line is driven high. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect.
8	D4D		DAT4 input/output signal data value
		Read 0	If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT4 line (low). If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 0.
		Write 0	If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT4 line is driven low. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect.
		Read 1	If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT4 line (high) If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 1.
		Write 1	If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT4 line is driven high. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect.
7	D3D		DAT3 input/output signal data value
		Read 0	If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT3 line (low). If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 0.
		Write 0	If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT3 line is driven low. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect.
		Read 1	If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT3 line (high) If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 1.
		Write 1	If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT3 line is driven high. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect.
6	D2D		DAT2 input/output signal data value
		Read 0	If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT2 line (low). If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 0.
		Write 0	If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT2 line is driven low. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect.
		Read 1	If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT2 line (high) If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 1.
		Write 1	If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT2 line is driven high. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 17-23. System Test Register (SD_SYSTEST) Field Descriptions (continued)

Bit	Field	Value	Description
5	D1D		DAT1 input/output signal data value
		Read 0	If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT1 line (low). If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 0.
		Write 0	If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT1 line is driven low. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect.
		Read 1	If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT1 line (high) If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 1.
		Write 1	If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT1 line is driven high. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect.
4	D0D		DAT0 input/output signal data value
		Read 0	If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT0 line (low). If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 0.
		Write 0	If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT0 line is driven low. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect.
		Read 1	If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), returns the value on the DAT0 line (high) If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), returns 1.
		Write 1	If SD_SYSTEST[3] DDIR bit = 0 (output mode direction), the DAT0 line is driven high. If SD_SYSTEST[3] DDIR bit = 1 (input mode direction), no effect.
3	DDIR		Control of the DAT[7:0] pins direction
		Read 0	No action. Returns 0.
		Write 0	The DAT lines are outputs (host to card).
		Read 1	No action. Returns 1.
		Write 1	The DAT lines are inputs (card to host).
2	CDAT		CMD input/output signal data value
		Read 0	If SD_SYSTEST[1] CDIR bit = 1 (input mode direction), returns the value on the CMD line (low). If SD_SYSTEST[1] CDIR bit = 0 (output mode direction), returns 0 .
		Write 0	If SD_SYSTEST[1] CDIR bit = 0 (output mode direction), the CMD line is driven low. If SD_SYSTEST[1] CDIR bit = 1 (input mode direction), no effect.
		Read 1	If SD_SYSTEST[1] CDIR bit = 1 (input mode direction), returns the value on the CMD line (high) If SD_SYSTEST[1] CDIR bit = 0 (output mode direction), returns 1 .
		Write 1	If SD_SYSTEST[1] CDIR bit = 0 (output mode direction), the CMD line is driven high. If SD_SYSTEST[1] CDIR bit = 1 (input mode direction), no effect.
1	CDIR		Control of the CMD pin direction
		Read 0	No action. Returns 0.
		Write 0	The CMD line is an output (host to card).
		Read 1	No action. Returns 1.
		Write 1	The CMD line is an input (card to host) .
0	MCKD		MMC clock output signal data value
		Read 0	No action. Returns 0.
		Write 0	The output clock is driven low.
		Read 1	No action. Returns 1.
		Write 1	The output clock is driven high.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.5.8 Configuration Register (SD_CON)

This register is used:

- To select the functional mode for any card
- To send an initialization sequence to any card
- To send an initialization sequence to any card
- To enable the detection on the mmc_dat[1] signal of a card interrupt for SDIO cards only

It also configures:

- The parameters related to the card detect and write protect input signals

The configuration register (SD_CON) is shown in Figure 17-44 and described in Table 17-24.

Figure 17-44. Configuration Register (SD_CON)

31							24
Reserved							
R-0							
23	22	21	20	19	18	17	16
Reserved		SDMA_LnE	DMA_MnS	DDR	BOOT_CF0	BOOT_ACK	CLKEXTFREE
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
15	14	13	12	11	10	9	8
PADEN	Reserved		CEATA	CTPL	DVAL		WPP
R/W-0	R-0		R/W-0	R/W-0	R/W-3h		R/W-0
7	6	5	4	3	2	1	0
CDP	MIT	DW8	MODE	STR	HR	INIT	OD
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

Table 17-24. Configuration Register (SD_CON) Field Descriptions

Bit	Field	Value	Description
31-22	Reserved	0	Reserved bit field. Do not write any value.
21	SDMA_LnE		Slave DMA Level/Edge Request. The waveform of the DMA request can be configured either edge sensitive with early de-assertion on first access to SD_DATA register or late de-assertion, request remains active until last allowed data written into SD_DATA.
		0	Slave DMA edge sensitive.
		1	Slave DMA level sensitive.
20	DMA_MnS		DMA Master or Slave selection. When this bit is set and the controller is configured to use the DMA, Ocp master interface is used to get datas from system using ADMA2 procedure (direct access to the memory). This option is only available if generic parameter MADMA_EN is asserted to 1.
		0	The controller is slave on data transfers with system.
		1	Not available on this device.
19	DDR		Dual Data Rate mode. When this register is set, the controller uses both clock edge to emit or receive data. Odd bytes are transmitted on falling edges and even bytes are transmitted on rise edges. It only applies on Data bytes and CRC, Start, end bits and CRC status are kept full cycle. This bit field is only meaningful and active for even clock divider ratio of SD_SYSCTL[CLKD], it is insensitive to SD_HCTL[HSPE] setting
		0	Standard modeData are transmitted on a single edge.
		1	Data Bytes and CRC are transmitted on both edges.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 17-24. Configuration Register (SD_CON) Field Descriptions (continued)

Bit	Field	Value	Description
18	BOOT_CF0		Boot Status Supported. This register is set when the CMD line needs to be forced to 0 for a boot sequence. CMD line is driven to 0 after writing in SD_CMD. The line is released when this bit field is de-asserted and aborts data transfer in case of a pending transaction.
		Read 0	CMD line not forced.
		Read 1	CMD line is released when it was previously forced to 0 by a boot sequence.
		Write 0	CMD line forced to 0 is enabled.
		Write 1	CMD line forced to 0 is enabled and will be active after writing into SD_CMD register.
17	BOOT_ACK		Book acknowledge received. When this bit is set the controller should receive a boot status on DAT0 line after next command issued. If no status is received a data timeout will be generated.
		0	No acknowledge to be received.
		1	A boot status will be received on DAT0 line after issuing a command.
16	CLKEXTFREE		External clock free running. This register is used to maintain card clock out of transfer transaction to enable slave module (for example to generate a synchronous interrupt on mmc_dat[1]). The Clock will be maintain only if SD_SYSCTL[2] CEN bit is set.
		0	External card clock is cut off outside active transaction period.
		1	External card clock is maintain even out of active transaction period only if SD_SYSCTL[2] CEN bit is set.
15	PADEN		Control power for MMC lines. This register is only useful when MMC PADs contain power saving mechanism to minimize its leakage power. It works as a GPIO that directly control the ACTIVE pin of PADs. Excepted for mmc_dat[1] , the signal is also combine outside the module with the dedicated power control SD_CON[11] CTPL bit.
		0	ADPIDLE module pin is not forced, it is automatically generated by the MMC fsms.
		1	ADPIDLE module pin is forced to active state
14-13	Reserved	0	Any writes to these bits will result in 0.
12	CEATA		CE-ATA control mode (MMC cards compliant with CE-ATA). This bit selects the active level of the out-of-band interrupt coming from MMC cards. The usage of the Out-of-Band signal (OBI) is not supported.
		0	Standard MMC/SD/SDIO mode.
		1	CE-ATA mode. Next commands are considered as CE-ATA commands.
11	CTPL		Control Power for mmc_dat[1] line (SD cards). By default, this bit is cleared to 0 and the host controller automatically disables all the input buffers outside of a transaction to minimize the leakage current.
			SDIO cards. When this bit is set to 1, the host controller automatically disables all the input buffers except the buffer of mmc_dat[1] outside of a transaction in order to detect asynchronous card interrupt on mmc_dat[1] line and minimize the leakage current of the buffers.
		0	Disable all the input buffers outside of a transaction.
		1	Disable all the input buffers except the buffer of mmc_dat[1] outside of a transaction.
10-9	DVAL		Debounce filter value (all cards). This register is used to define a debounce period to filter the card detect input signal (SDCD). The usage of the card detect input signal (SDCD) is optional and depends on the system integration and the type of the connector housing that accommodates the card.
		0	33 us debounce period
		1h	231 us debounce period
		2h	1 ms debounce period
		3h	8.4 ms debounce period
8	WPP		Write protect polarity (SD and SDIO cards only). This bit selects the active level of the write protect input signal (SDWP). The usage of the write protect input signal (SDWP) is optional and depends on the system integration and the type of the connector housing that accommodates the card.
		0	Active high level
		1	Active low level

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 17-24. Configuration Register (SD_CON) Field Descriptions (continued)

Bit	Field	Value	Description
7	CDP		Card detect polarity (all cards). This bit selects the active level of the write protect input signal (SDWP). The usage of the write protect input signal (SDWP) is optional and depends on the system integration and the type of the connector housing that accommodates the card.
		0	Active high level
		1	Active low level
6	MIT		MMC interrupt command (MMC cards only). This bit must be set to 1, when the next write access to the command register (SD_CMD) is for writing a MMC interrupt command (CMD40) requiring the command timeout detection to be disabled for the command response.
		0	Command timeout enabled.
		1	Command timeout disabled.
5	DW8		8-bit mode MMC select (MMC cards only). For SD/SDIO cards, this bit must be cleared to 0. For MMC card, this bit must be set following a valid SWITCH command (CMD6) with the correct value and extend CSD index written in the argument. Prior to this command, the MMC card configuration register (CSD and EXT_CSD) must be verified for compliancy with MMC standard specification.
		0	1-bit or 4-bit data width
		1	8-bit data width
4	MODE		Mode select (all cards). This bit selects the functional mode.
		0	Functional mode. Transfers to the MMC/SD/SDIO cards follow the card protocol. The MMC clock is enabled. MMC/SD transfers are operated under the control of the SD_CMD register.
		1	SYSTEST mode. SYSTEST mode. The signal pins are configured as general-purpose input/output and the 1024-byte buffer is configured as a stack memory accessible only by the local host or system DMA. The pins retain their default type (input, output or in-out). SYSTEST mode is operated under the control of the SYSTEST register.
3	STR		Stream command (MMC cards only). This bit must be set to 1 only for the stream data transfers (read or write) of the adtc commands. Stream read is a class 1 command (CMD11READ_DAT_UNTIL_STOP). Stream write is a class 3 command (CMD20WRITE_DAT_UNTIL_STOP).
		0	Block oriented data transfer
		1	Stream oriented data transfer
2	HR		Broadcast host response (MMC cards only). This register is used to force the host to generate a 48-bit response for bc command type. It can be used to terminate the interrupt mode by generating a CMD40 response by the core. In order to have the host response to be generated in open drain mode, the register SD_CON[OD] must be set to 1. When SD_CON[12] CEATA bit is set to 1 and SD_ARG cleared to 0, when writing 0000 0000h into SD_CMD register, the host controller performs a 'command completion signal disable' token (i.e., mmc_cmd line held to 0 during 47 cycles followed by a 1).
		0	The host does not generate a 48-bit response instead of a command.
		1	The host generates a 48-bit response instead of a command or a command completion signal disable token.
1	INIT		Send initialization stream (all cards). When this bit is set to 1, and the card is idle, an initialization sequence is sent to the card. An initialization sequence consists of setting the mmc_cmd line to 1 during 80 clock cycles. The initialization sequence is mandatory - but it is not required to do it through this bit - this bit makes it easier. Clock divider (SD_SYSCTL[15:6] CLKD bits) should be set to ensure that 80 clock periods are greater than 1ms. Notein this mode, there is no command sent to the card and no response is expected. A command complete interrupt will be generated once the initialization sequence is completed. SD_STAT[0] CC bit can be polled.
		0	The host does not send an initialization sequence
		1	The host sends an initialization sequence
0	OD		Card open drain mode (MMC cards only). This bit must be set to 1 for MMC card commands 1, 2, 3 and 40, and if the MMC card bus is operating in open-drain mode during the response phase to the command sent. Typically, during card identification mode when the card is either in idle, ready or ident state. It is also necessary to set this bit to 1, for a broadcast host response (see Broadcast host response register SD_CON[2] HR bit)
		0	No open drain
		1	Open drain or broadcast host response

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.5.9 Power Counter Register (SD_PWCNT)

This register is used to program a mmc counter to delay command transfers after activating the PAD power, this value depends on PAD characteristics and voltage. The power counter register (SD_PWCNT) is shown in Figure 17-45 and described in Table 17-25.

Figure 17-45. Power Counter Register (SD_PWCNT)

31–16	15–0
Reserved	PWRCNT
R-0	R/W-0

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

Table 17-25. Power Counter Register (SD_PWCNT) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	These bits are initialized to zero, and writes to them are ignored. Reads return 0.
15-0	PWRCNT		Power counter register. This register is used to introduce a delay between the PAD ACTIVE pin assertion and the command issued.
		0	No additional delay added
		1h	TCF delay (card clock period)
		2h	TCF x 2 delay (card clock period)
		FFFEh	TCF x 65534 delay (card clock period)
		FFFFh	TCF x 65535 delay (card clock period)

17.5.10 SDMA System Address (SD_SDMASA)

This register is used to program a mmc counter to delay command transfers after activating the PAD power. This value depends on PAD characteristics and voltage. The SDMA System address (SD_SDMASA) is shown in Figure 17-42 and described in Table 17-22.

Figure 17-46. Card Status Response Error (SD_SDMASA)

31–0
SDMA_SYSADDR
R-0

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

Table 17-26. Card Status Response Error (SD_SDMASA) Field Descriptions

Bit	Field	Value	Description
31-0	SDMA_SYSADDR		This register contains the system memory address for a SDMA transfer. When the Host Controller stops a SDMA transfer, this register shall point to the system address of the next contiguous data position. It can be accessed only if no transaction is executing (i.e., after a transaction has stopped). Read operations during transfers may return an invalid value. The Host Driver shall initialize this register before starting a SDMA transaction. After SDMA has stopped, the next system address of the next contiguous data position can be read from this register. The SDMA transfer waits at the every boundary specified by the Host SDMA Buffer Boundary in the Block Size register. The Host Controller generates DMA Interrupt to request the Host Driver to update this register. The Host Driver sets the next system address of the next data position to this register. When the most upper byte of this register (003h) is written, the Host Controller restarts the SDMA transfer. When restarting SDMA by the Resume command or by setting Continue Request in the Block Gap Control register, the Host Controller shall start at the next contiguous address stored here in the SDMA System Address register. ADMA does not use this register.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.5.11 Transfer Length Configuration Register (SD_BLK)

This register shall be used for any card. SD_BLK[BLEN] is the block size register. SD_BLK[NBLK] is the block count register.

The transfer length configuration register (SD_BLK) is shown in Figure 17-47 and described in Table 17-27.

Figure 17-47. Transfer Length Configuration Register (SD_BLK)

31–16	15–12	11–0
NBLK	Reserved	BLEN
R/W-0	R-0	R/W-0

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

Table 17-27. Transfer Length Configuration Register (SD_BLK) Field Descriptions

Bit	Field	Value	Description
31-16	NBLK		Blocks count for current transfer. This register is enabled when Block count Enable (SD_CMD[1] BCE bit) is set to 1 and is valid only for multiple block transfers. Setting the block count to 0 results no data blocks being transferred. NoteThe host controller decrements the block count after each block transfer and stops when the count reaches zero. This register can be accessed only if no transaction is executing (i.e., after a transaction has stopped). Read operations during transfers may return an invalid value and write operation will be ignored. In suspend context, the number of blocks yet to be transferred can be determined by reading this register. When restoring transfer context prior to issuing a Resume command, The local host shall restore the previously saved block count.
		0	Stop count
		1h	1 block
		2h	2 blocks
		FFFFh	65535 blocks
15-12	Reserved	0	Reserved bit field. Do not write any value.
11-0	BLEN		Transfer block size. This register specifies the block size for block data transfers. Read operations during transfers may return an invalid value, and write operations are ignored. When a CMD12 command is issued to stop the transfer, a read of the BLEN field after transfer completion (SD_STAT[1] TC bit set to 1) will not return the true byte number of data length while the stop occurs but the value written in this register before transfer is launched.
		0	No data transfer
		1h	1 byte block length
		2h	2 bytes block length
		3h	3 bytes block length
		1FFh	511 bytes block length
		200h	512 bytes block length
		7FFh	2047 bytes block length
		800h	2048 bytes block length

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.5.12 Command Argument Register (SD_ARG)

This register contains command argument specified as bit 39-8 of Command-Format. These registers must be initialized prior to sending the command itself to the card (write action into the register SD_CMD register). Only exception is for a command index specifying stuff bits in arguments, making a write unnecessary.

The command argument register (SD_ARG) is shown in Figure 17-48 and described in Table 17-28.

Figure 17-48. Command Argument Register (SD_ARG)

31	0
ARG	
R/W-0	

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

Table 17-28. Command Argument Register (SD_ARG) Field Descriptions

Bit	Field	Value	Description
31-0	ARG		Command argument bits [31:0] .

17.5.13 Command and Transfer Mode Register (SD_CMD)

- SD_CMD[31:16] = the command register
- SD_CMD[15:0] = the transfer mode

This register configures the data and command transfers. A write into the most significant byte send the command. A write into SD_CMD[15:0] registers during data transfer has no effect.

This register can be used for any card.

NOTE: In SYSTEST mode, a write into SD_CMD register will not start a transfer.

The command and transfer mode register (SD_CMD) is shown in Figure 17-49 and described in Table 17-29.

Figure 17-49. Command and Transfer Mode Register (SD_CMD)

31	30	29					24
Reserved		INDX					
R-0		R/W-0					

23	22	21	20	19	18	17	16
CMD_TYPE		DP	CICE	CCCE	Reserved	RSP_TYPE	
R/W-0		R/W-0	R/W-0	R/W-0	R-0	R/W-0	

15	8
Reserved	
R-0	

7	6	5	4	3	2	1	0
Reserved		MSBS	DDIR	Reserved	ACEN	BCE	DE
R-0		R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 17-29. Command and Transfer Mode Register (SD_CMD) Field Descriptions

Bit	Field	Value	Description
31-30	Reserved	0	Reserved bit field. Do not write any value.
29-24	INDX		Command index binary encoded value from 0 to 63 specifying the command number send to card.
		0	CMD0 or ACMD0
		1h	CMD1 or ACMD1
		2h	CMD2 or ACMD2
		3h	CMD3 or ACMD3
		4h	CMD4 or ACMD4
		5h	CMD5 or ACMD5
		6h	CMD6 or ACMD6
		7h	CMD7 or ACMD7
		8h	CMD8 or ACMD8
		9h	CMD9 or ACMD9
		Ah	CMD10 or ACMD10
		Bh	CMD11 or ACMD11
		Ch	CMD12 or ACMD12
		Dh	CMD13 or ACMD13
		Eh	CMD14 or ACMD14
		Fh	CMD15 or ACMD15
		10h	CMD16 or ACMD16
		11h	CMD17 or ACMD17
		12h	CMD18 or ACMD18
		13h	CMD19 or ACMD19
		14h	CMD20 or ACMD20
		15h	CMD21 or ACMD21
		16h	CMD22 or ACMD22
		17h	CMD23 or ACMD23
		18h	CMD24 or ACMD24
		19h	CMD25 or ACMD25
		1Ah	CMD26 or ACMD26
		1Bh	CMD27 or ACMD27
		1Ch	CMD28 or ACMD28
		1Dh	CMD29 or ACMD29
		1Eh	CMD30 or ACMD30
		1Fh	CMD31 or ACMD31
		20h	CMD32 or ACMD32
		21h	CMD33 or ACMD33
		22h	CMD34 or ACMD34
		23h	CMD35 or ACMD35
		24h	CMD36 or ACMD36
		25h	CMD37 or ACMD37
		26h	CMD38 or ACMD38
		27h	CMD39 or ACMD39
		28h	CMD40 or ACMD40
		29h	CMD41 or ACMD41
		2Ah	CMD42 or ACMD42
		2Bh	CMD43 or ACMD43
		2Ch	CMD44 or ACMD44
		2Dh	CMD45 or ACMD45

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 17-29. Command and Transfer Mode Register (SD_CMD) Field Descriptions (continued)

Bit	Field	Value	Description
		2Eh	CMD46 or ACMD46
		2Fh	CMD47 or ACMD47
		30h	CMD48 or ACMD48
		31h	CMD49 or ACMD49
		32h	CMD50 or ACMD5
		33h	CMD51 or ACMD5
		34h	CMD52 or ACMD5
		35h	CMD53 or ACMD5
		36h	CMD54 or ACMD5
		37h	CMD55 or ACMD5
		38h	CMD56 or ACMD5
		39h	CMD57 or ACMD5
		3Ah	CMD58 or ACMD5
		3Bh	CMD59 or ACMD5
		3Ch	CMD60 or ACMD60
		3Dh	CMD61 or ACMD61
		3Eh	CMD62 or ACMD62
		3Fh	CMD63 or ACMD63
23-22	CMD_TYPE		Command type. This register specifies three types of special commandSuspend, Resume and Abort. These bits shall be cleared to 0b00 for all other commands.
		0	Others commands
		1h	Upon CMD52 "Bus Suspend" operation
		2h	Upon CMD52 "Function Select" operation
		3h	Upon CMD12 or CMD52 "I/O Abort" command
21	DP		Data present select. This register indicates that data is present and mmc_dat line shall be used. It must be cleared to 0 in the following conditions: • Command using only mmc_cmd line • Command with no data transfer but using busy signal on mmc_dat0 • Resume command
		0	Command with no data transfer
		1	Command with data transfer
20	CICE		Command Index check enable. This bit must be set to 1 to enable index check on command response to compare the index field in the response against the index of the command. If the index is not the same in the response as in the command, it is reported as a command index error (SD_STAT[19] CIE bit set to1) NoteThe CICE bit cannot be configured for an Auto CMD12, then index check is automatically checked when this command is issued.
		0	Index check disable
		1	Index check enable
19	CCCE		Command CRC check enable. This bit must be set to 1 to enable CRC7 check on command response to protect the response against transmission errors on the bus. If an error is detected, it is reported as a command CRC error (SD_STAT[17] CCRC bit set to 1). NoteThe CCCE bit cannot be configured for an Auto CMD12, and then CRC check is automatically checked when this command is issued.
		0	CRC7 check disable
		1	CRC7 check enable
18	Reserved	0	Reserved bit field. Do not write any value.
17-16	RSP_TYPE		Response type. This bits defines the response type of the command.
		0	No response
		1h	Response Length 136 bits
		2h	Response Length 48 bits
		3h	Response Length 48 bits with busy after response

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 17-29. Command and Transfer Mode Register (SD_CMD) Field Descriptions (continued)

Bit	Field	Value	Description
15-6	Reserved	0	Reserved bit field. Do not write any value.
5	MSBS		Multi/Single block select. This bit must be set to 1 for data transfer in case of multi block command. For any others command this bit shall be cleared to 0.
		0	Single block. If this bit is 0, it is not necessary to set the register SD_BLK[31:16] NBLK bits.
		1	Multi block. When Block Count is disabled (SD_CMD[1] BCE bit is cleared to 0) in Multiple block transfers (SD_CMD[5] MSBS bit is set to 1), the module can perform infinite transfer.
4	DDIR		Data transfer Direction. Select This bit defines either data transfer will be a read or a write.
		0	Data Write (host to card)
		1	Data Read (card to host)
3	Reserved	0	Reserved bit field. Do not write any value.
2	ACEN		Auto CMD12 Enable (SD cards only). When this bit is set to 1, the host controller issues a CMD12 automatically after the transfer completion of the last block. The Host Driver shall not set this bit to issue commands that do not require CMD12 to stop data transfer. In particular, secure commands do not require CMD12. For CE-ATA commands (SD_CON[12] CEATA bit set to 1), auto CMD12 is useless; therefore when this bit is set the mechanism to detect command completion signal, named CCS, interrupt is activated.
		0	Auto CMD12 disable
		1	Auto CMD12 enable or CCS detection enabled.
1	BCE		Block Count Enable (Multiple block transfers only). This bit is used to enable the block count register (SD_BLK[31:16] NBLK bits). When Block Count is disabled (SD_CMD[1] BCE bit is cleared to 0) in Multiple block transfers (SD_CMD[5] MSBS bits is set to 1), the module can perform infinite transfer.
		0	Block count disabled for infinite transfer.
		1	Block count enabled for multiple block transfer with known number of blocks
0	DE		DMA Enable. This bit is used to enable DMA mode for host data access.
		0	DMA mode disable
		1	DMA mode enable

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.5.14 Command Response[31:0] Register (SD_RSP10)

This 32-bit register holds bits positions [31:0] of command response type R1, R1b, R2, R3, R4, R5, R5b, or R6. The command response[31:0] register (SD_RSP10) is shown in Figure 17-50 and described in Table 17-30.

Figure 17-50. Command Response[31:0] Register (SD_RSP10)

31 – 16	15 – 0
RSP1	RSP0
R-0	R-0

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

Table 17-30. Command Response[31:0] Register (SD_RSP10) Field Descriptions

Bit	Field	Value	Description
31-16	RSP1		Command Response [31:16]
15-0	RSP0		Command Response [15:0]

17.5.15 Command Response[63:32] Register (SD_RSP32)

This 32-bit register holds bits positions [63:32] of command response type R2. The command response[63:32] register (SD_RSP32) is shown in Figure 17-51 and described in Figure 17-51.

Figure 17-51. Command Response[63:32] Register (SD_RSP32)

31 – 16	15 – 0
RSP3	RSP2
R-0	R-0

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

Table 17-31. Command Response[63:32] Register (SD_RSP32) Field Descriptions

Bit	Field	Value	Description
31-16	RSP3		Command Response [63:48]
15-0	RSP2		Command Response [47:32]

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.5.16 Command Response[95:64] Register (SD_RSP54)

This 32-bit register holds bits positions [95:64] of command response type R2. The command response[95:64] register (SD_RSP54) is shown in Figure 17-52 and described in Table 17-32.

Figure 17-52. Command Response[95:64] Register (SD_RSP54)

31 16	15 0
RSP5	RSP4
R-0	R-0

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

Table 17-32. Command Response[95:64] Register (SD_RSP54) Field Descriptions

Bit	Field	Value	Description
31-16	RSP5		Command Response [95:80]
15-0	RSP4		Command Response [79:64]

17.5.17 Command Response[127:96] Register (SD_RSP76)

This 32-bit register holds bits positions [127:96] of command response type R2. The command response[127:96] register (SD_RSP76) is shown in Figure 17-53 and described in Table 17-33.

Figure 17-53. Command Response[127:96] Register (SD_RSP76)

31 16	15 0
RSP7	RSP6
R-0	R-0

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

Table 17-33. Command Response[127:96] Register (SD_RSP76) Field Descriptions

Bit	Field	Value	Description
31-16	RSP7		Command Response [127:112]
15-0	RSP6		Command Response [111:96]

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.5.18 Data Register (SD_DATA)

This register is the 32-bit entry point of the buffer for read or write data transfers. The buffer size is 32bitsx256(1024 bytes). Bytes within a word are stored and read in little endian format. This buffer can be used as two 512 byte buffers to transfer data efficiently without reducing the throughput.

Sequential and contiguous access is necessary to increment the pointer correctly. Random or skipped access is not allowed. In little endian, if the local host accesses this register byte-wise or 16bit-wise, the least significant byte (bits [7:0]) must always be written/read first. The update of the buffer address is done on the most significant byte write for full 32-bit DATA register or on the most significant byte of the last word of block transfer.

Example 1Byte or 16-bit access

- Mbyteen[3:0]=0001 (1-byte) => Mbyteen[3:0]=0010 (1-byte) => Mbyteen[3:0]=1100 (2-bytes) OK
- Mbyteen[3:0]=0001 (1-byte) => Mbyteen[3:0]=0010 (1-byte) => Mbyteen[3:0]=0100 (1-byte) OK
- Mbyteen[3:0]=0001 (1-byte) => Mbyteen[3:0]=0010 (1-byte) => Mbyteen[3:0]=1000 (1-byte) Bad

The data register (SD_DATA) is shown in Figure 17-54 and described in Table 17-34.

Figure 17-54. Data Register (SD_DATA)

31	0
DATA	
R/W-0	

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

Table 17-34. Data Register (SD_DATA) Field Descriptions

Bit	Field	Value	Description
31-0	DATA		Data register [31:0]. In functional mode (SD_CON[4] MODE bit set to the default value 0): A read access to this register is allowed only when the buffer read enable status is set to 1 (SD_PSTATE[11] BRE bit), otherwise a bad access (SD_STAT[29] BADA bit) is signaled. A write access to this register is allowed only when the buffer write enable status is set to 1 (SD_PSTATE[10] BWE bit), otherwise a bad access (SD_STAT[29] BADA bit) is signaled and the data is not written.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.5.19 Present State Register (SD_PSTATE)

The Host can get the status of the Host controller from this 32-bit read only register. The present state register (SD_PSTATE) is shown in Figure 17-55 and described in Table 17-35.

Figure 17-55. Present State Register (SD_PSTATE)

31–25	24	23–20	19	18	17	16
Reserved	CLEV	DLEV	WP	CDPL	CSS	CINS
R-0	R-x	R-x	R-0	R-0	R-0	R-0

15–12	11	10	9	8	7–3	2	1	0
Reserved	BRE	BWE	RTA	WTA	Reserved	DLA	DATI	CMDI
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

Table 17-35. Present State Register (SD_PSTATE) Field Descriptions

Bit	Field	Value	Description
31-25	Reserved	0	Reserved bit field. Do not write any value.
24	CLEV		mmc_cmd line signal level. This status is used to check the mmc_cmd line level to recover from errors, and for debugging. The value of this register after reset depends on the mmc_cmd line level at that time.
		Read 0	The mmc_cmd line level is 0.
		Read 1	The mmc_cmd line level is 1.
23-20	DLEV		mmc_dat[3:0] line signal level • mmc_dat3 => bit 23 • mmc_dat2 => bit 22 • mmc_dat1 => bit 21 • mmc_dat0 => bit 20 This status is used to check mmc_dat line level to recover from errors, and for debugging. This is especially useful in detecting the busy signal level from mmc_dat0 . The value of these registers after reset depends on the mmc_dat lines level at that time.
19	WP		Write Protect. MMC/SD/SDIO1 only. SDIO cards only. This bit reflects the write protect input pin (SDWP) level. The value of this register after reset depends one the protect input pin (SDWP) level at that time.
		Read 0	If SD_CON[8] WPP is cleared to 0 (default), the card is write protected, otherwise the card is not write protected.
		Read 1	If SD_CON[8] WPP is cleared to 0 (default), the card is not write protected, otherwise the card is write protected.
18			Card Detect Pin Level. MMC/SD/SDIO1 only. SDIO cards only. This bit reflects the inverse value of the card detect input pin (SDCD). Debouncing is not performed on this bit and is valid only when Card State is stable. (SD_PSTATE[17] is set to 1). This bit must be debounced by software. The value of this register after reset depends on the card detect input pin (SDCD) level at that time.
		Read 0	The value of the card detect input pin (SDCD) is 1.
		Read 1	The value of the card detect input pin (SDCD) is 0.
17			Card State Stable. This bit is used for testing. It is set to 1 only when Card Detect Pin Level is stable (SD_PSTATE[18] CPDL). Debouncing is performed on the card detect input pin (SDCD) to detect card stability. This bit is not affected by software reset.
		Read 0	Reset or Debouncing.
		Read 1	Reset or Debouncing.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 17-35. Present State Register (SD_PSTATE) Field Descriptions (continued)

Bit	Field	Value	Description
16			Card inserted. This bit is the debounced value of the card detect input pin (SDCD). An inactive to active transition of the card detect input pin (SDCD) will generate a card insertion interrupt (SD_STAT[CINS]). A active to inactive transition of the card detect input pin (SDCD) will generate a card removal interrupt (SD_STAT[REM]). This bit is not affected by a software reset.
		Read 0	If SD_CON[CDP] is cleared to 0 (default), no card is detected. The card may have been removed from the card slot. If SD_CON[CDP] is set to 1, the card has been inserted.
		Read 1	If SD_CON[CDP] is cleared to 0 (default), the card has been inserted from the card slot. If SD_CON[CDP] is set to 1, no card is detected. The card may have been removed from the card slot.
15-12	Reserved	0	Reserved bit field. Do not write any value.
11	BRE		Buffer read enable. This bit is used for non-DMA read transfers. It indicates that a complete block specified by SD_BLK[10:0] BLEN bits has been written in the buffer and is ready to be read. It is cleared to 0 when the entire block is read from the buffer. It is set to 1 when a block data is ready in the buffer and generates the Buffer read ready status of interrupt (SD_STAT[5] BRR bit).
		Read 0	Read BLEN bytes disable
		Read 1	Read BLEN bytes enable. Readable data exists in the buffer.
10	BWE		Buffer Write enable. This status is used for non-DMA write transfers. It indicates if space is available for write data.
		Read 0	There is no room left in the buffer to write BLEN bytes of data.
		Read 1	There is enough space in the buffer to write BLEN bytes of data.
9	RTA		Read transfer active. This status is used for detecting completion of a read transfer. It is set to 1 after the end bit of read command or by activating a continue request (SD_HCTL[17] CR bit) following a stop at block gap request. This bit is cleared to 0 when all data have been read by the local host after last block or after a stop at block gap request.
		Read 0	No valid data on the mmc_dat lines.
		Read 1	Read data transfer on going.
8	WTA		Write transfer active. This status indicates a write transfer active. It is set to 1 after the end bit of write command or by activating a continue request (SD_HCTL[17] CR bit) following a stop at block gap request. This bit is cleared to 0 when CRC status has been received after last block or after a stop at block gap request.
		Read 0	No valid data on the mmc_dat lines.
		Read 1	Write data transfer on going.
7-3	Reserved	0	Reserved bit field. Do not write any value.
2	DLA		mmc_dat line active. This status bit indicates whether one of the mmc_dat lines is in use. In the case of read transactions (card to host)This bit is set to 1 after the end bit of read command or by activating continue request SD_HCTL[17] CR bit. This bit is cleared to 0 when the host controller received the end bit of the last data block or at the beginning of the read wait mode. In the case of write transactions (host to card)This bit is set to 1 after the end bit of write command or by activating continue request SD_HCTL[17] CR bit. This bit is cleared to 0 on the end of busy event for the last block; host controller must wait 8 clock cycles with line not busy to really consider not "busy state" or after the busy block as a result of a stop at gap request.
		Read 0	mmc_dat line inactive
		Read 1	mmc_dat line active
1	DATI		Command inhibit (mmc_dat). This status bit is generated if either mmc_dat line is active (SD_PSTATE[2] DLA bit) or Read transfer is active (SD_PSTATE[9] RTA bit) or when a command with busy is issued. This bit prevents the local host to issue a command. A change of this bit from 1 to 0 generates a transfer complete interrupt (SD_STAT[1] TC bit).
		Read 0	Issuing of command using the mmc_dat lines is allowed
		Read 1	Issuing of command using mmc_dat lines is not allowed

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 17-35. Present State Register (SD_PSTATE) Field Descriptions (continued)

Bit	Field	Value	Description
0	CMDI		Command inhibit(mmc_cmd). This status bit indicates that the mmc_cmd line is in use. This bit is cleared to 0 when the most significant byte is written into the command register. This bit is not set when Auto CMD12 is transmitted. This bit is cleared to 0 in either the following cases: • After the end bit of the command response, excepted if there is a command conflict error (SD_STAT[17] CCRC bit or SD_STAT[18] CEB bit set to 1) or a Auto CMD12 is not executed (SD_AC12[0] ACNE bit). • After the end bit of the command without response (SD_CMD[17:16] RSP_TYPE bits set to "00"). In case of a command data error is detected (SD_STAT[19] CTO bit set to 1), this register is not automatically cleared.
		Read 0	Issuing of command using mmc_cmd line is allowed
		Read 1	Issuing of command using mmc_cmd line is not allowed

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.5.20 Control Register (SD_HCTL)

This register defines the host controls to set power, wake-up and transfer parameters.

- SD_HCTL[31:24] = Wake-up control
- SD_HCTL[23:16] = Block gap control
- SD_HCTL[15:8] = Power control
- SD_HCTL[7:0] = Host control

NOTE: If your device does not support MMC cards, then those bits in this register which are meant for MMC card use should be assumed to be reserved.

The control register (SD_HCTL) is shown in Figure 17-56 and described in Table 17-36.

Figure 17-56. Control Register (SD_HCTL)

31–28	27	26	25	24	23–20	19	18	17	16
Reserved	OBWE	REM	INS	IWE	Reserved	IBG	RWC	CR	SBGR
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0

15–12	11–9	8	7	6	5	4–3	2	1	0
Reserved	SDVS	SDBP	CDSS	CDTL	Rsvd	DMAS	HSPE	DTW	Rsvd
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R/W-2h	R/W-0	R/W-0	R-0

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

Table 17-36. Control Register (SD_HCTL) Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Reserved bit field. Do not write any value.
27	OBWE		Wake-up event enable for 'out-of-band' Interrupt. This bit enables wake-up events for 'out-of-band' assertion. Wake-up is generated if the wake-up feature is enabled (SD_SYSCONFIG[2] ENAWAKEUP bit). The write to this register is ignored when SD_CON[14] OBIE bit is not set.
		0	Disable wake-up on 'out-of-band' Interrupt
		1	Enable wake-up on 'out-of-band' Interrupt
26	REM		Wake-up event enable on SD card removal. This bit enables wake-up events for card removal assertion. Wake-up is generated if the wake-up feature is enabled (SD_SYSCONFIG[2] ENAWAKEUP bit).
		0	Disable wake-up on card removal
		1	Enable wake-up on card removal
25	INS		Wake-up event enable on SD card insertion This bit enables wake-up events for card insertion assertion. Wake-up is generated if the wake-up feature is enabled (SD_SYSCONFIG[2] ENAWAKEUP bit).
		0	Disable wake-up on card insertion
		1	Enable wake-up on card insertion
24	IWE		Wake-up event enable on SD card interrupt. This bit enables wake-up events for card interrupt assertion. Wake-up is generated if the wake-up feature is enabled (SD_SYSCONFIG[2] ENAWAKEUP bit) and enable status bit is set (SD_IE[8] CIRQ_ENABLE bit).
		0	Disable wake-up on card interrupt
		1	Enable wake-up on card interrupt
23-20	Reserved	0	Reserved bit field. Do not write any value.
19	IBG		Interrupt block at gap. This bit is valid only in 4-bit mode of SDIO card to enable interrupt detection in the interrupt cycle at block gap for a multiple block transfer. For MMC cards and for SD card this bit should be cleared to 0.
		0	Disable interrupt detection at the block gap in 4-bit mode
		1	Enable interrupt detection at the block gap in 4-bit mode

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 17-36. Control Register (SD_HCTL) Field Descriptions (continued)

Bit	Field	Value	Description
18	RWC		Read wait control. The read wait function is optional only for SDIO cards. If the card supports read wait, this bit must be enabled, then requesting a stop at block gap (SD_HCTL[16] SBGR bit) generates a read wait period after the current end of block. Be careful, if read wait is not supported it may cause a conflict on mmc_dat line.
		0	Disable read wait control. Suspend/resume cannot be supported.
		1	Enable read wait control
17	CR		Continue request. This bit is used to restart a transaction that was stopped by requesting a stop at block gap (SD_HCTL[16] SBGR bit). Set this bit to 1 restarts the transfer. The bit is automatically cleared to 0 by the host controller when transfer has restarted, that is, mmc_dat line is active (SD_PSTATE[2] DLA bit) or transferring data (SD_PSTATE[8] WTA bit). The Stop at block gap request must be disabled (SD_HCTL[16] SBGR bit =0) before setting this bit.
		0	No affect
		1	Transfer restart
16	SBGR		Stop at block gap request. This bit is used to stop executing a transaction at the next block gap. The transfer can restart with a continue request (SD_HCTL[17] CR bit) or during a suspend/resume sequence. In case of read transfer, the card must support read wait control. In case of write transfer, the host driver shall set this bit after all block data written. Until the transfer completion (SD_STAT[1] TC bit set to 1), the host driver shall leave this bit set to 1.If this bit is set, the local host shall not write to the data register (SD_DATA).
		0	Transfer mode
		1	Stop at block gap
15-12	Reserved	0	Reserved bit field. Do not write any value.
11-9	SDVS		SD bus voltage select (All cards). The host driver should set these bits to select the voltage level for the card according to the voltage supported by the system (SD_CAPA[26] VS18 bit, SD_CAPA[25] VS30 bit, SD_CAPA[24] VS33 bit) before starting a transfer.
			MMCHS2. This field must be set to 5h.
			MMCHS3. This field must be set to 5h.
		5h	1.8 V (Typical)
		6h	3.0 V (Typical)
		7h	3.3 V (Typical)
8	SDBP		SD bus power. Before setting this bit, the host driver shall select the SD bus voltage (SD_HCTL[11:9] SDVS bits). If the host controller detects the No card state, this bit is automatically cleared to 0. If the module is power off, a write in the command register (SD_CMD) will not start the transfer. A write to this bit has no effect if the selected SD bus voltage is not supported according to capability register (SD_CAPA[VS*]).
		0	Power off
		1	Power on
7	CDSS		Card Detect Signal Selection. This bit selects source for the card detection. When the source for the card detection is switched, the interrupt should be disabled during the switching period by clearing the Interrupt Status/Signal Enable register in order to mask unexpected interrupt being caused by the glitch. The Interrupt Status/Signal Enable should be disabled during over the period of debouncing.
		0	SDCD# is selected (for normal use).
		1	The Card Detect Test Level is selected (for test purposes).
6	CDTL		Card Detect Test Level. This bit is enabled while the Card Detect Signal Selection is set to 1 and it indicates card inserted or not.
		0	No card
		1	Card inserted.
5	Reserved	0	Reserved bit field. Do not write any value.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 17-36. Control Register (SD_HCTL) Field Descriptions (continued)

Bit	Field	Value	Description
4-3	DMAS		DMA Select. One of the supported DMA modes can be selected. The host driver shall check support of DMA modes by referencing the Capabilities register. Use of selected DMA is determined by DMA Enable of the Transfer Mode register. This register is only meaningful when MADMA_EN is set to 1. When MADMA_EN is cleared to 0 the bit field is read only and returned value is 0.
		0-1h	Reserved
		2h	32-bit Address ADMA2 is selected.
		3h	Reserved
2	HSPE		High Speed Enable. Before setting this bit, the Host Driver shall check the High Speed Support in the Capabilities register. If this bit is cleared to 0 (default), the Host Controller outputs CMD line and DAT lines at the falling edge of the SD Clock. If this bit is set to 1, the Host Controller outputs CMD line and DAT lines at the rising edge of the SD Clock. This bit shall not be set when dual data rate mode is activated in SD_CON[DDR].
		0	Normal speed mode
		1	High speed mode
1	DTW		Data transfer width. This bit must be set following a valid SET_BUS_WIDTH command (ACMD6) with the value written in bit 1 of the argument. Prior to this command, the SD card configuration register (SCR) must be verified for the supported bus width by the SD card.
		0	1-bit Data width (mmc_dat0 used)
		1	4-bit Data width (mmc_dat[3:0] used)
0	Reserved	0	Reserved bit field. Do not write any value.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.5.21 SD System Control Register (SD_SYSCTL)

This register defines the system controls to set software resets, clock frequency management and data timeout.

- SD_SYSCTL[31:24] = Software resets
- SD_SYSCTL[23:16] = Timeout control
- SD_SYSCTL[15:0] = Clock control

The SD system control register (SD_SYSCTL) is shown in Figure 17-57 and described in Table 17-37.

Figure 17-57. SD System Control Register (SD_SYSCTL)

31–27	26	25	24	23–20	19–16
Reserved	SRD	SRC	SRA	Reserved	DTO
R-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0

15–6	5–3	2	1	0
CLKD	Reserved	CEN	ICS	ICE
R/W-0	R-0	R/W-0	R-0	R/W-0

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

Table 17-37. SD System Control Register (SD_SYSCTL) Field Descriptions

Bit	Field	Value	Description
31-27	Reserved	0	Reserved bit field. Do not write any value.
26	SRD		Software reset for mmc_dat line. This bit is set to 1 for reset and released to 0 when completed. mmc_dat finite state machine in both clock domain are also reset. These registers are cleared by the SD_SYSCTL[26] SRD bit: • SD_DATA • SD_PSTATEBRE, BWE, RTA, WTA, DLA and DATI • SD_HCTLSBGR and CR • SD_STATBRR, BWR, BGE and TC Interconnect and MMC buffer data management is reinitialized. **Note**If a soft reset is issued when an interrupt is asserted, data may be lost.
		0	Reset completed
		1	Software reset for mmc_dat line
25	SRC		Software reset for mmc_cmd line. This bit is set to 1 for reset and released to 0 when completed. mmc_cmd finite state machine in both clock domain are also reset. These registers are cleared by the SD_SYSCTL[25] SRC bit: • SD_PSTATECMDI • SD_STATCC Interconnect and MMC command status management is reinitialized. **Note**If a soft reset is issued when an interrupt is asserted, data may be lost.
		0	Reset completed
		1	Software reset for mmc_cmd line
24	SRA		Software reset for all. This bit is set to 1 for reset , and released to 0 when completed. This reset affects the entire host controller except for the card detection circuit and capabilities registers.
		0	Reset completed
		1	Software reset for all the design
23-20	Reserved	0	Reserved bit field. Do not write any value.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 17-37. SD System Control Register (SD_SYSCTL) Field Descriptions (continued)

Bit	Field	Value	Description
19-16	DTO		Data timeout counter value and busy timeout. This value determines the interval by which mmc_dat lines timeouts are detected. The host driver needs to set this bit field based on: • The maximum read access time (NAC) (Refer to the SD Specification Part1 Physical Layer) • The data read access time values (TAAC and NSAC) in the card specific data register (CSD) of the card • The timeout clock base frequency (SD_CAPA[5:0] TCF bits) If the card does not respond within the specified number of cycles, a data timeout error occurs (SD_STAT[20] DTO bit). The SD_SYSCTL[19,16] DTO bit field is also used to check busy duration, to generate busy timeout for commands with busy response or for busy programming during a write command. Timeout on CRC status is generated if no CRC token is present after a block write.
		0	TCF x 2^13
		1h	TCF x 2^14
		Eh	TCF x 2^27
		Fh	Reserved
15-6	CLKD		Clock frequency select. These bits define the ratio between a reference clock frequency (system dependant) and the output clock frequency on the mmc_clk pin of either the memory card (MMC, SD, or SDIO).
		0	Clock Ref bypass
		1h	Clock Ref bypass
		2h	Clock Ref / 2
		3h	Clock Ref / 3
		3FFh	Clock Ref / 1023
5-3	Reserved	0	Reserved bit field. Do not write any value.
2	CEN		Clock enable. This bit controls if the clock is provided to the card or not.
		0	The clock is not provided to the card . Clock frequency can be changed .
		1	The clock is provided to the card and can be automatically gated when SD_SYSCONFIG[0] AUTOIDLE bit is set to 1 (default value) . The host driver shall wait to set this bit to 1 until the Internal clock is stable (SD_SYSCTL[1] ICS bit).
1	ICS		Internal clock stable (status)This bit indicates either the internal clock is stable or not.
		0	The internal clock is not stable.
		1	The internal clock is stable after enabling the clock (SD_SYSCTL[0] ICE bit) or after changing the clock ratio (SD_SYSCTL[15:6] CLKD bits).
0	ICE		Internal clock enable. This register controls the internal clock activity. In very low power state, the internal clock is stopped. NoteThe activity of the debounce clock (used for wake-up events) and the interface clock (used for reads and writes to the module register map) are not affected by this register.
		0	The internal clock is stopped (very low power state).
		1	The internal clock oscillates and can be automatically gated when SD_SYSCONFIG[0] AUTOIDLE bit is set to 1 (default value).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.5.22 Interrupt Status Register (SD_STAT)

The interrupt status regroups all the status of the module internal events that can generate an interrupt.

- SD_STAT[31:16] = Error Interrupt Status
- SD_STAT[15:0] = Normal Interrupt Status

The error bits are located in the upper 16 bits of the SD_STAT register. All bits are cleared by writing a 1 to them. Additionally, bits 15 and 8 serve as special error bits. These cannot be cleared by writing a 1 to them. Bit 15 (ERRI) is automatically cleared when the error causing to ERRI to be set is handled. (that is, when bits 31:16 are cleared, bit 15 will be automatically cleared). Bit 8 (CIRQ) is cleared by writing a 0 to SD_IE[8] (masking the interrupt) and servicing the interrupt.

The interrupt status register (SD_STAT) is shown in Figure 17-58 and described in Table 17-38.

Figure 17-58. Interrupt Status Register (SD_STAT)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		BADA	CERR	Reserved		ADMAE	ACE	Rsvd	DEB	DCRC	DTO	CIE	CEB	CCRC	CTO
R-0		R/W-0	R/W-0	R-0		R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

15	14	10	9	8	7	6	5	4	3	2	1	0
ERRI	Reserved	BSR	OBI	CIRQ	CREM	CINS	BRR	BWR	DMA	BGE	TC	CC
R-0	R-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

Table 17-38. Interrupt Status Register (SD_STAT) Field Descriptions

Bit	Field	Value	Description
31-30	Reserved	0	Reserved bit field. Do not write any value
29	BADA		Bad access to data space. This bit is set automatically to indicate a bad access to buffer when not allowed: • During a read access to the data register (SD_DATA) while buffer reads are not allowed (SD_PSTATE[11] BRE bit =0) • During a write access to the data register (SD_DATA) while buffer writes are not allowed (SD_PSTATE[10] BWE bit=0)
		Read 0	No interrupt
		Write 0	Status bit unchanged
		Read 1	Bad access
		Write 1	Status is cleared.
28	CERR		Card error. This bit is set automatically when there is at least one error in a response of type R1, R1b, R6, R5 or R5b. Only bits referenced as type E (error) in status field in the response can set a card status error. An error bit in the response is flagged only if corresponding bit in card status response error SD_CSRE in set. There is no card error detection for autoCMD12 command. The host driver shall read SD_RSP76 register to detect error bits in the command response.
		Read 0	No error
		Write 0	Status bit unchanged
		Read 1	Card error
		Write 1	Status is cleared.
27-26	Reserved	0	Reserved bit field. Do not write any value

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 17-38. Interrupt Status Register (SD_STAT) Field Descriptions (continued)

Bit	Field	Value	Description
25	ADMAE		ADMA Error. This bit is set when the Host Controller detects errors during ADMA based data transfer. The state of the ADMA at an error occurrence is saved in the ADMA Error Status Register. In addition, the Host Controller generates this interrupt when it detects invalid descriptor data (Valid=0) at the ST_FDS state. ADMA Error State in the ADMA Error Status indicates that an error occurs in ST_FDS state. The Host Driver may find that Valid bit is not set at the error descriptor.
		Read 0	No interrupt
		Write 0	Status bit unchanged
		Read 1	ADMA error
		Write 1	Status is cleared.
24	ACE		Auto CMD12 error. This bit is set automatically when one of the bits in Auto CMD12 Error status register has changed from 0 to 1.
		Read 0	No error
		Write 0	Status bit unchanged
		Read 1	AutoCMD12 error
		Write 1	Status is cleared.
23	Reserved	0	Reserved bit field. Do not write any value
22	DEB		Data End Bit error. This bit is set automatically when detecting a 0 at the end bit position of read data on mmc_dat line or at the end position of the CRC status in write mode.
		Read 0	No error
		Write 0	Status bit unchanged
		Read 1	Data end bit error
		Write 1	Status is cleared.
21	DCRC		Data CRC Error. This bit is set automatically when there is a CRC16 error in the data phase response following a block read command or if there is a 3-bit CRC status different of a position "010" token during a block write command.
		Read 0	No error
		Write 0	Status bit unchanged
		Read 1	Data CRC error
		Write 1	Status is cleared.
20	DTO		Data timeout error. This bit is set automatically according to the following conditions: • Busy timeout for R1b, R5b response type • Busy timeout after write CRC status • Write CRC status timeout • Read data timeout
		Read 0	No error
		Write 0	Status bit unchanged
		Read 1	Time out
		Write 1	Status is cleared.
19	CIE		Command index error. This bit is set automatically when response index differs from corresponding command index previously emitted. It depends on the enable bit (SD_CMD[20] CICE).
		Read 0	No error
		Write 0	Status bit unchanged
		Read 1	Command index error
		Write 1	Status is cleared.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 17-38. Interrupt Status Register (SD_STAT) Field Descriptions (continued)

Bit	Field	Value	Description
18	CEB		Command end bit error. This bit is set automatically when detecting a 0 at the end bit position of a command response.
		Read 0	No error
		Write 0	Status bit unchanged
		Read 1	Command end bit error
		Write 1	Status is cleared.
17	CCRC		Command CRC error. This bit is set automatically when there is a CRC7 error in the command response depending on the enable bit (SD_CMD[19] CCCE).
		Read 0	No error
		Write 0	Status bit unchanged
		Read 1	Command CRC error
		Write 1	Status is cleared.
16	CTO		Command timeout error. This bit is set automatically when no response is received within 64 clock cycles from the end bit of the command.
			For commands that reply within 5 clock cycles - the timeout is still detected at 64 clock cycles.
		Read 0	No error
		Write 0	Status bit unchanged
		Read 1	Time Out
		Write 1	Status is cleared.
15	ERRI		Error interrupt. If any of the bits in the Error Interrupt Status register (SD_STAT[31:16]) are set, then this bit is set to 1. Therefore the host driver can efficiently test for an error by checking this bit first.
			Writes to this bit are ignored.
		Read 0	No interrupt
		Read 1	Error interrupt event(s) occurred
14-11	Reserved	0	Reserved bit field. Do not write any value
10	BSR		Boot Status Received Interrupt. This bit is set automatically when SD_CON[BOOT] is set 1 or 2 and a boot status is received on DAT[0] line. This interrupt is only useful for MMC card.
		Read 0	No interrupt
		Write 0	Status bit unchanged
		Read 1	Boot Status Received Interrupt occurred.
		Write 1	Status is cleared.
9	OBI		Out-of-band interrupt (This interrupt is only useful for MMC card). This bit is set automatically when SD_CON[14] OBIE bit is set and an out-of-band interrupt occurs on OBI pin.
			The interrupt detection depends on polarity controlled by SD_CON[13] OBIP bit.
			The out-of-band interrupt signal is a system specific feature for future use, this signal is not required for existing specification implementation.
		Read 0	No out-of-band interrupt
		Write 0	Status bit unchanged
		Read 1	Interrupt out-of-band occurs
		Write 1	Status is cleared.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 17-38. Interrupt Status Register (SD_STAT) Field Descriptions (continued)

Bit	Field	Value	Description
8	CIRQ		Card interrupt. This bit is only used for SD and SDIO cards. In 1-bit mode, interrupt source is asynchronous (can be a source of asynchronous wake-up). In 4-bit mode, interrupt source is sampled during the interrupt cycle. In CE-ATA mode, interrupt source is detected when the card drives mmc_cmd line to zero during one cycle after data transmission end. All modes above are fully exclusive. The controller interrupt must be clear by setting SD_IE[8] CIRQ_ENABLE to 0, then the host driver must start the interrupt service with card (clearing card interrupt status) to remove card interrupt source. Otherwise the Controller interrupt will be reasserted as soon as SD_IE[8] CIRQ_ENABLE is set to 1. Writes to this bit are ignored.
		Read 0	No card interrupt
		Read 1	Generate card interrupt
7	CREM		Card Removal. This bit is set automatically when SD_PSTATE[CINS] changes from 1 to 0. A clear of this bit doesn't affect Card inserted present state (SD_PSTATE[CINS]).
		Read 0	Card State stable or debouncing
		Write 0	Status bit unchanged
		Read 1	Card Removed
		Write 1	Status is cleared
6	CINS		Card Insertion. This bit is set automatically when SD_PSTATE[CINS] changes from 0 to 1. A clear of this bit doesn't affect Card inserted present state (SD_PSTATE[CINS]).
		Read 0	Card State stable or debouncing
		Write 0	Status bit unchanged
		Read 1	Card inserted
		Write 1	Status is cleared.
5	BRR		Buffer read ready. This bit is set automatically during a read operation to the card (see class 2 - block oriented read commands) when one block specified by the SD_BLK[10:0] BLEN bit field is completely written in the buffer. It indicates that the memory card has filled out the buffer and that the local host needs to empty the buffer by reading it. NoteIf the DMA receive-mode is enabled, this bit is never set; instead a DMA receive request to the main DMA controller of the system is generated.
		Read 0	Not ready to read buffer
		Write 0	Status bit unchanged
		Read 1	Ready to read buffer
		Write 1	Status is cleared.
4	BWR		Buffer write ready. This bit is set automatically during a write operation to the card (see class 4 - block oriented write command) when the host can write a complete block as specified by SD_BLK[10:0] BLEN. It indicates that the memory card has emptied one block from the buffer and that the local host is able to write one block of data into the buffer. NoteIf the DMA transmit mode is enabled, this bit is never set; instead, a DMA transmit request to the main DMA controller of the system is generated.
		Read 0	Not ready to write buffer
		Write 0	Status bit unchanged
		Read 1	Ready to write buffer
		Write 1	Status is cleared.
3	DMA		DMA Interrupt. This status is set when an interrupt is required in the ADMA instruction and after the data transfer completion.
		Read 0	DMA Interrupt detected
		Write 0	Status bit unchanged
		Read 1	No DMA Interrupt
		Write 1	Status is cleared.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 17-38. Interrupt Status Register (SD_STAT) Field Descriptions (continued)

Bit	Field	Value	Description
2	BGE		Block gap event. When a stop at block gap is requested (SD_HCTL[16] SBGR bit), this bit is automatically set when transaction is stopped at the block gap during a read or write operation. This event does not occur when the stop at block gap is requested on the last block. In read mode, a 1-to-0 transition of the mmc_dat line active status (SD_PSTATE[2] DLA bit) between data blocks generates a Block gap event interrupt.
		Read 0	No block gap event
		Write 0	Status bit unchanged
		Read 1	Transaction stopped at block gap
		Write 1	Status is cleared
1	TC		Transfer completed. This bit is always set when a read/write transfer is completed or between two blocks when the transfer is stopped due to a stop at block gap request (SD_HCTL[16] SBGR bit). This bit is also set when exiting a command in a busy state (if the command has a busy notification capability). In Read modeThis bit is automatically set on completion of a read transfer (SD_PSTATE[9] RTA bit). In write modeThis bit is set automatically on completion of the mmc_dat line use (SD_PSTATE[2] DLA bit).
		Read 0	No transfer complete
		Write 0	Status bit unchanged
		Read 1	Data transfer complete
		Write 1	Status is cleared
0	CC		Command complete. This bit is set when a 1-to-0 transition occurs in the register command inhibit (SD_PSTATE[0] CMDI bit) If the command is a type for which no response is expected, then the command complete interrupt is generated at the end of the command. A command timeout error (SD_STAT[16] CTO bit) has higher priority than command complete (SD_STAT[0] CC bit). If a response is expected but none is received, then a command timeout error is detected and signaled instead of the command complete interrupt.
		Read 0	No command complete
		Write 0	Status bit unchanged
		Read 1	Command complete
		Write 1	Status is cleared

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.5.23 Interrupt SD Enable Register (SD_IE)

This register allows to enable/disable the module to set status bits, on an event-by-event basis.

- SD_IE[31:16] = Error Interrupt Status Enable
- SD_IE[15:0] = Normal Interrupt Status Enable

The interrupt SD enable register (SD_IE) is shown in Figure 17-59 and described in Table 17-39.

Figure 17-59. Interrupt SD Enable Register (SD_IE)

31	30	29	28	27	26	25	24
Reserved		BADA_ENABLE	CERR_ENABLE	Reserved		ADMA_ENABLE	ACE_ENABLE
R-0		R/W-0	R/W-0	R-0		R/W-0	R/W-0

23	22	21	20	19	18	17	16
Reserved	DEB_ENABLE	DCRC_ENABLE	DTO_ENABLE	CIE_ENABLE	CEB_ENABLE	CCRC_ENABLE	CTO_ENABLE
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

15	14	11	10	9	8
NULL	Reserved		BSR_ENABLE	OBI_ENABLE	CIRQ_ENABLE
R-0	R-0		R/W-0	R/W-0	R/W-0

7	6	5	4	3	2	1	0
CREM_ENABLE	CINS_ENABLE	BRR_ENABLE	BWR_ENABLE	DMA_ENABLE	BGE_ENABLE	TC_ENABLE	CC_ENABLE
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

Table 17-39. Interrupt SD Enable Register (SD_IE) Field Descriptions

Bit	Field	Value	Description
31-30	Reserved	0	Reserved bit field. Do not write any value.
29	BADA_ENABLE		Bad access to data space interrupt enable
		0	Masked
		1	Enabled
28	CERR_ENABLE		Card error interrupt enable
		0	Masked
		1	Enabled
27-26	Reserved	0	Reserved bit field. Do not write any value.
25	ADMA_ENABLE		ADMA error Interrupt Enable
		0	Masked
		1	Enabled
24	ACE_ENABLE		Auto CMD12 error interrupt enable
		0	Masked
		1	Enabled
23	Reserved	0	Reserved bit field. Do not write any value.
22	DEB_ENABLE		Data end bit error interrupt enable
		0	Masked
		1	Enabled
21	DCRC_ENABLE		Data CRC error interrupt enable
		0	Masked
		1	Enabled
20	DTO_ENABLE		Data timeout error interrupt enable
		0	Masked
		1	Enabled

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 17-39. Interrupt SD Enable Register (SD_IE) Field Descriptions (continued)

Bit	Field	Value	Description
19	CIE_ENABLE		Command index error interrupt enable
		0	Masked
		1	Enabled
18	CEB_ENABLE		Command end bit error interrupt enable
		0	Masked
		1	Enabled
17	CCRC_ENABLE		Command CRC error interrupt enable
		0	Masked
		1	Enabled
16	CTO_ENABLE		Command timeout error interrupt enable
		0	Masked
		1	Enabled
15	NULL	0	Fixed to 0. The host driver shall control error interrupts using the Error Interrupt Signal Enable register. Writes to this bit are ignored.
14-11	Reserved	0	Reserved bit field. Do not write any value.
10	BSR_ENABLE		Boot Status Interrupt Enable A write to this register when SD_CON[BOOT] is cleared to 0 is ignored.
		0	Masked
		1	Enabled
9	OBI_ENABLE		Out-of-band interrupt enable A write to this register when SD_CON[14] OBIE is cleared to 0 is ignored.
		0	Masked
		1	Enabled
8	CIRQ_ENABLE		Card interrupt enable. A clear of this bit also clears the corresponding status bit. During 1-bit mode, if the interrupt routine does not remove the source of a card interrupt in the SDIO card, the status bit is reasserted when this bit is set to 1. This bit must be set to 1 when entering in smart idle mode to enable system to identity wake-up event and to allow controller to clear internal wake-up source.
		0	Masked
		1	Enabled
7	CREM_ENABLE		Card Removal interrupt Enable This bit must be set to 1 when entering in smart idle mode to enable system to identity wake-up event and to allow controller to clear internal wake-up source.
		0	Masked
		1	Enabled
6	CINS_ENABLE		Card Insertion interrupt Enable This bit must be set to 1 when entering in smart idle mode to enable system to identity wake-up event and to allow controller to clear internal wake-up source.
		0	Masked
		1	Enabled
5	BRR_ENABLE		Buffer read ready interrupt enable
		0	Masked
		1	Enabled
4	BWR_ENABLE		Buffer write ready interrupt enable
		0	Masked
		1	Enabled
3	DMA_ENABLE		DMA interrupt enable
		0	Masked
		1	Enable
2	BGE_ENABLE		Block gap event interrupt enable
		0	Masked
		1	Enabled

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 17-39. Interrupt SD Enable Register (SD_IE) Field Descriptions (continued)

Bit	Field	Value	Description
1	TC_ENABLE		Transfer completed interrupt enable
		0	Masked
		1	Enabled
0	CC_ENABLE		Command completed interrupt enable
		0	Masked
		1	Enabled

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.5.24 Interrupt Signal Enable Register (SD_ISE)

This register allows you to enable/disable the module to set status bits, on an event-by-event basis.

- SD_ISE[31:16] = Error Interrupt Signal Enable
- SD_ISE[15:0] = Normal Interrupt Signal Enable

The Interrupt Signal Enable Register (SD_ISE) is shown in Figure 17-60 and described in Table 17-40.

Figure 17-60. Interrupt Signal Enable Register (SD_ISE)

31	30	29	28	27	26	25	24
Reserved		BADA_SIGEN	CERR_SIGEN	Reserved		ADMA_SIGEN	ACE_SIGEN
R-0		R/W-0	R/W-0	R-0		R/W-0	R/W-0

23	22	21	20	19	18	17	16
Reserved	DEB_SIGEN	DCRC_SIGEN	DTO_SIGEN	CIE_SIGEN	CEB_SIGEN	CCRC_SIGEN	CTO_SIGEN
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

15	14	11	10	9	8
NULL	Reserved		BSR_SIGEN	OBI_SIGEN	CIRQ_SIGEN
R-0	R-0		R/W-0	R/W-0	R/W-0

7	6	5	4	3	2	1	0
CREM_SIGEN	CINS_SIGEN	BRR_SIGEN	BWR_SIGEN	DMA_SIGEN	BGE_SIGEN	TC_SIGEN	CC_SIGEN
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

Table 17-40. Interrupt Signal Enable Register (SD_ISE) Field Descriptions

Bit	Field	Value	Description
31-30	Reserved	0	Reserved bit field. Do not write any value.
29	BADA_SIGEN		Bad access to data space interrupt enable
		0	Masked
		1	Enabled
28	CERR_SIGEN		Card error interrupt signal status enable
		0	Masked
		1	Enabled
27-26	Reserved	0	Reserved bit field. Do not write any value.
25	ADMA_SIGEN		ADMA error signal status enable
		0	Masked
		1	Enabled
24	ACE_SIGEN		Auto CMD12 error signal status enable
		0	Masked
		1	Enabled
23	Reserved	0	Reserved bit field. Do not write any value.
22	DEB_SIGEN		Data end bit error signal status enable
		0	Masked
		1	Enabled
21	DCRC_SIGEN		Data CRC error signal status enable
		0	Masked
		1	Enabled
20	DTO_SIGEN		Data timeout error signal status enable
		0	Masked
			The host controller provides the clock to the card until the card sends the data or the transfer is aborted.
		1	Enabled

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 17-40. Interrupt Signal Enable Register (SD_ISE) Field Descriptions (continued)

Bit	Field	Value	Description
19	CIE_SIGEN		Command index error signal status enable
		0	Masked
		1	Enabled
18	CEB_SIGEN		Command end bit error signal status enable
		0	Masked
		1	Enabled
17	CCRC_SIGEN		Command CRC error signal status enable
		0	Masked
		1	Enabled
16	CTO_SIGEN		Command timeout error signal status enable
		0	Masked
		1	Enabled
15	NULL		Fixed to 0. The host driver shall control error interrupts using the error interrupt signal enable register. Writes to this bit are ignored.
14-11	Reserved	0	Reserved bit field. Do not write any value.
10	BSR_SIGEN		Boot Status signal status enable. A write to this register when SD_CON[BOOT] is cleared to 0 is ignored
		0	Masked
		1	Enabled
9	OBI_SIGEN		Out-of-band interrupt signal status enable. A write to this register when SD_CON[14] OBIE is cleared to 0 is ignored.
		0	Masked
		1	Enabled
8	CIRQ_SIGEN		Card interrupt signal status enable. A clear of this bit also clears the corresponding status bit. During 1-bit mode, if the interrupt routine does not remove the source of a card interrupt in the SDIO card, the status bit is reasserted when this bit is set to 1. This bit must be set to 1 when entering in smart idle mode to enable system to identity wake-up event and to allow controller to clear internal wake-up source.
		0	Masked
		1	Enabled
7	CREM_SIGEN		Card Removal signal status enable This bit must be set to 1 when entering in smart idle mode to enable system to identity wake-up event and to allow controller to clear internal wake-up source.
		0	Masked
		1	Enabled
6	CINS_SIGEN		Card Insertion signal status enable. This bit must be set to 1 when entering in smart idle mode to enable system to identity wake-up event and to allow controller to clear internal wake-up source.
		0	Masked
		1	Enabled
5	BRR_SIGEN		Buffer read ready signal status enable
		0	Masked
		1	Enabled
4	BWR_SIGEN		Buffer write ready signal status enable
		0	Masked
		1	Enabled
3	DMA_SIGEN		DMA signal status enable
		0	Masked
		1	Enabled
2	BGE_SIGEN		Block gap event signal status enable
		0	Masked
		1	Enabled

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 17-40. Interrupt Signal Enable Register (SD_ISE) Field Descriptions (continued)

Bit	Field	Value	Description
1	TC_SIGEN		Transfer completed signal status enable
		0	Masked
		1	Enabled
0	CC_SIGEN		Command completed signal status enable
		0	Masked
		1	Enabled

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.5.25 Auto CMD12 Error Status Register (SD_AC12)

The host driver may determine which of the errors cases related to Auto CMD12 has occurred by checking this SD_AC12 register when an auto CMD12 error interrupt occurs.

This register is valid only when auto CMD12 is enabled (SD_CMD[2] ACEN bit) and auto CMD12Error (SD_STAT[24] ACE bit) is set to 1.

NOTE: These bits are automatically reset when starting a new adtc command with data.

The auto CMD12 error status register (SD_AC12) is shown in Figure 17-61 and described in Table 17-41.

Figure 17-61. Auto CMD12 Error Status Register (SD_AC12)

31	8	7	6	5	4	3	2	1	0
Reserved		CNI	Reserved		ACIE	ACEB	ACCE	ACTO	ACNE
R-0		R-0	R-0		R-0	R-0	R-0	R-0	R-0

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

Table 17-41. Auto CMD12 Error Status Register (SD_AC12) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved bit field. Do not write any value.
7	CNI		Command not issue by auto CMD12 error. If this bit is set to 1, it means that pending command is not executed due to auto CMD12 error ACEB, ACCE, ACTO, or ACNE.
		Read 0	Not error
		Read 1	Command not issued
6-5	Reserved	0	Reserved bit field. Do not write any value.
4	ACIE		Auto CMD12 index error. This bit is a set to 1 when response index differs from corresponding command auto CMD12 index previously emitted.
			This bit depends on the command index check enable (SD_CMD[20] CICE bit).
		Read 0	No error
		Read 1	Auto CMD12 index error
3	ACEB		Auto CMD12 end bit error. This bit is set to 1 when detecting a 0 at the end bit position of auto CMD12 command response.
		Read 0	No error
		Read 1	AutoCMD12 end bit error
2	ACCE		Auto CMD12 CRC error. This bit is automatically set to 1 when a CRC7 error is detected in the auto CMD12 command response depending on the enable in the SD_CMD[19] CCCE bit.
		Read 0	No error
		Read 1	Auto CMD12 CRC error
1	ACTO		Auto CMD12 timeout error. This bit is set to 1 if no response is received within 64 clock cycles from the end bit of the auto CMD12 command.
		Read 0	No error
		Read 1	Auto CMD12 time out
0	ACNE		Auto CMD12 not executed. This bit is set to 1 if multiple block data transfer command has started and if an error occurs in command before auto CMD12 starts.
		Read 0	Auto CMD12 executed
		Read 1	Auto CMD12 not executed

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.5.26 Capabilities Register (SD_CAPA)

This register lists the capabilities of the MMC/SD/SDIO host controller. The capabilities register (SD_CAPA) is shown in Figure 17-62 and described in Table 17-42.

Figure 17-62. Capabilities Register (SD_CAPA)

31–29	28	27	26	25	24	23	22	21	20	19	18	17–16
Reserved	64BIT	Rsvd	VS18	VS30	VS33	SRS	DS	HSS	Rsvd	AD2S	Rsvd	MBL
R-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R-1	R-1	R-1	R-0	R-1	R-0	R-1h

15–14	13–8	7	6	5–0
Reserved	BCF	TCU	Rsvd	TCF
R-0	R-0	R-1	R-0	R-0

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

Table 17-42. Capabilities Register (SD_CAPA) Field Descriptions

Bit	Field	Value	Description
31-29	Reserved	0	Reserved bit field. Do not write any value.
28	64BIT		64 Bit System Bus Support. Setting 1 to this bit indicates that the Host Controller supports 64-bit address descriptor mode and is connected to 64-bit address system bus.
		Read 0	32 bit System bus address
		Read 1	64 bit System bus address
27	Reserved	0	Reserved bit. Any write to this bit results in 0.
26	VS18		Voltage support 1.8 V. Initialization of this register (via a write access to this register) depends on the system capabilities. The host driver shall not modify this register after the initialization. This register is only reinitialized by a hard reset (via mmc_RESET signal).
			Note. The device supports both 1.8v and 3.3v cards. However, MMC/SD/SDIO0 and MMC/SD/SDIO1 must be configured to the same voltage. You cannot configure MMC/SD/SDIO0, for example, to be 1.8v and MMC/SD/SDIO1 to be 3.3v. MMC/SD/SDIO2 is independent of the other two and can be set to either voltage regardless of what the other two are set to.
		Read 0	1.8 V not supported
		Write 0	1.8 V not supported
		Read 1	1.8 V supported
		Write 1	1.8 V supported
25	VS30		Voltage support 3.0V. Initialization of this register (via a write access to this register) depends on the system capabilities. The host driver shall not modify this register after the initialization. This register is only reinitialized by a hard reset (via mmc_RESET signal)
			Note. The device supports both 1.8v and 3.3v cards, however MMC/SD/SDIO0 and MMC/SD/SDIO1 must be configured to the same voltage. You cannot configure MMC/SD/SDIO0, for example, to be 1.8v and MMC/SD/SDIO1 to be 3.3v. MMC/SD/SDIO2 is independent of the other two and can be set to either voltage regardless of what the other two are set to.
		Read 0	3.0 V not supported
		Write 0	3.0 V not supported
		Read 1	3.0 V supported
		Write 1	3.0 V supported
24	VS33		Voltage support 3.3V. Initialization of this register (via a write access to this register) depends on the system capabilities. The host driver shall not modify this register after the initialization. This register is only reinitialized by a hard reset (via mmc_RESET signal)
		Read 0	3.3 V not supported
		Write 0	3.3 V not supported
		Read 1	3.3 V supported
		Write 1	3.3 V supported

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 17-42. Capabilities Register (SD_CAPA) Field Descriptions (continued)

Bit	Field	Value	Description
23	SRS		Suspend/resume support (SDIO cards only). This bit indicates whether the host controller supports Suspend/Resume functionality.
		Read 0	The Host controller does not suspend/resume functionality.
		Read 1	The Host controller supports suspend/resume functionality.
22	DS		DMA support. This bit indicates that the Host controller is able to use DMA to transfer data between system memory and the Host controller directly.
		Read 0	DMA not supported
		Read 1	DMA supported
21	HSS		High-speed support. This bit indicates that the host controller supports high speed operations and can supply an up-to-52 MHz clock to the card.
		Read 0	DMA not supported
		Read 1	DMA supported
20	Reserved	0	Reserved bit field. Do not write any value.
19	AD2S		This bit indicates whether the Host Controller is capable of using ADMA2. It depends on setting of generic parameter MADMA_EN.
		Read 0	ADMA2 supported
		Read 1	ADMA2 not supported
18	Reserved	0	Reserved bit field. Do not write any value.
17-16	MBL		Maximum block length. This value indicates the maximum block size that the host driver can read and write to the buffer in the host controller.
			The host controller supports 512 bytes and 1024 bytes block transfers.
		Read 0	512 bytes
		Read 1	1024 bytes
		Read 2	2048 bytes
15-14	Reserved	0	Reserved bit field. Do not write any value.
13-8	BCF		Base clock frequency for clock provided to the card.
		Read 0	The value indicating the base (maximum) frequency for the output clock provided to the card is system dependent and is not available in this register. Get the information via another method. See the *Power, Reset, and Clock Management* chapter for more information on the value of FUNC_96M_CLK clock signal.
7	TCU		Timeout clock unit. This bit shows the unit of base clock frequency used to detect Data Timeout Error (SD_STAT[20] DTO bit).
		Read 0	kHz
		Read 1	MHz
6	Reserved	0	Reserved. This bit is initialized to zero, and writes to it are ignored.
5-0	TCF		Timeout clock frequency. The timeout clock frequency is used to detect Data Timeout Error (SD_STAT[20] DTO bit).
		Read 0	The timeout clock frequency depends on the frequency of the clock provided to the card. The value of the timeout clock frequency is not available in this register.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.5.27 Maximum Current Capabilities Register (SD_CUR_CAPA)

This register indicates the maximum current capability for each voltage. The value is meaningful if the voltage support is set in the capabilities register (SD_CAPA).

Initialization of this register (via a write access to this register) depends on the system capabilities. The host driver shall not modify this register after the initialization.

This register is only reinitialized by a hard reset (via mmc_RESET signal). The maximum current capabilities register (SD_CUR_CAPA) is shown in Figure 17-63 and described in Table 17-43.

Figure 17-63. Maximum Current Capabilities Register (SD_CUR_CAPA)

31–24	23–16	15–8	7–0
Reserved	CUR_1V8	CUR_3V0	CUR_3V3
R-0	R/W-0	R/W-0	R/W-0

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

Table 17-43. Maximum Current Capabilities Register (SD_CUR_CAPA) Field Descriptions

Bit	Field	Value	Description
31-24	Reserved	0	Reserved. This bit is initialized to zero, and writes to it are ignored.
23-16	CUR_1V8		Maximum current for 1.8 V
		Read 0	The maximum current capability for this voltage is not available. Feature not implemented.
15-8	CUR_3V0		Maximum current for 3.0 V
		Read 0	The maximum current capability for this voltage is not available. Feature not implemented.
7-0	CUR_3V3		Maximum current for 3.3 V
		Read 0	The maximum current capability for this voltage is not available. Feature not implemented.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.5.28 Force Event Register for Error Interrupt Status (SD_FE)

The Force Event register is not a physically implemented register. Rather, it is an address at which the Error Interrupt Status register can be written. The effect of a write to this address will be reflected in the Error Interrupt Status Register, if corresponding bit of the Error Interrupt Status Enable Register is set.

The force event register (SD_FE) is shown in Figure 17-64 and described in Table 17-44.

Figure 17-64. Interrupt Signal Enable Register (SD_ISE)

31	30	29	28	27	26	25	24
Reserved		FE_BADA	FE_CERR	Reserved		FE_ADMAE	FE_ACE
R-0		W-0	W-0	R-0		W-0	W-0

23	22	21	20	19	18	17	16
Reserved	FE_DEB	FE_DCRC	FE_DTO	FE_CIE	FE_CEB	FE_CCRC	FE_CTO
R-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0

15	8
Reserved	
R-0	

7	6	5	4	3	2	1	0
FE_CNI	Reserved		FE_ACIE	FE_ACEB	FE_ACCE	FE_ACTO	FE_ACNE
W-0	R-0		W-0	W-0	W-0	W-0	W-0

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

Table 17-44. Force Event Register (SD_FE) Field Descriptions

Bits	Field	Value	Description
31-30	Reserved	0	Reserved bit field. Any writes to these bits result in 0.
29	FE_BADA		Force Event Bad access to data space.
		Write 0	No effect; no interrupt
		Write 1	Interrupt forced.
28	FE_CERR		Force Event Card error
		Write 0	No effect; no interrupt
		Write 1	Interrupt forced.
27-26	Reserved	0	Reserved bit field. Any writes to these bits result in 0.
25	FE_ADMAE		Force Event ADMA error
		Write 0	No effect; no interrupt
		Write 1	Interrupt forced.
24	FE_ACE		Force Event Auto CMD12 error.
		Write 0	No effect; no interrupt
		Write 1	Interrupt forced.
23	Reserved	0	Reserved bit field. Any writes to this bit result in 0.
22	FE_DEB		Force Event Data End Bit error.
		Write 0	No effect; no interrupt
		Write 1	Interrupt forced.
21	FE_DCRC		Force Event Data CRC error
		Write 0	No effect; no interrupt
		Write 1	Interrupt forced.
20	FE_DTO		Force Event Data timeout error
		Write 0	No effect; no interrupt
		Write 1	Interrupt forced.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 17-44. Force Event Register (SD_FE) Field Descriptions (continued)

Bits	Field	Value	Description
19	FE_CIE		Force Event Command index error
		Write 0	No effect; no interrupt
		Write 1	Interrupt forced.
18	FE_CEB		Force Event Command end bit error
		Write 0	No effect; no interrupt
		Write 1	Interrupt forced.
17	FE_CCRC		Force Event Comemand CRC error
		Write 0	No effect; no interrupt
		Write 1	Interrupt forced.
16	FE_CTO		Force Event Command Timeout error
		Write 0	No effect; no interrupt
		Write 1	Interrupt forced.
15-8	Reserved	0	Reserved bit field. Any writes to these bits result in 0.
7	FE_CNI		Force Event Command not issue by Auto CMD12 error
		Write 0	No effect; no interrupt
		Write 1	Interrupt forced.
6-5	Reserved	0	Reserved bit field. Any writes to these bits result in 0.
4	FE_ACIE		Force Event Auto CMD12 index error
		Write 0	No effect; no interrupt
		Write 1	Interrupt forced.
3	FE_ACEB		Force Event Auto CMD12 end bit error
		Write 0	No effect; no interrupt
		Write 1	Interrupt forced.
2	FE_ACCE		Force Event Auto CMD12 CRC error
		Write 0	No effect; no interrupt
		Write 1	Interrupt forced.
1	FE_ACTO		Force Event Auto CMD12 timeout error
		Write 0	No effect; no interrupt
		Write 1	Interrupt forced.
0	FE_ACNE		Force Event Auto CMD12 not executed.
		Write 0	No effect; no interrupt
		Write 1	Interrupt forced.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.5.29 ADMA Error Status Register (SD_ADMAES)

When an ADMA Error Interrupt has occurred, the ADMA Error States field in this register holds the ADMA state and the ADMA System Address Register holds the address around the error descriptor. For recovering the error, the Host Driver requires the ADMA state to identify the error descriptor address as follows:

- ST_STOP: Previous location set in the ADMA System Address register is the error descriptor address.
- ST_FDS: Current location set in the ADMA System Address register is the error descriptor address.
- ST_CADR: This sate is never set because do not generate ADMA error in this state.
- ST_TFR: Previous location set in the ADMA System Address register is the error descriptor address.

In the case of a write operation, the Host Driver should use ACMD22 to get the number of written block rather than using this information, since unwritten data may exist in the Host Controller. The Host Controller generates the ADMA Error Interrupt when it detects invalid descriptor data (Valid = 0) at the ST_FDS state. In this case, ADMA Error State indicates that an error occurs at ST_FDS state. The Host Driver may find that the Valid bit is not set in the error descriptor.

The ADMA error status register (SD_BLK) is shown in Figure 17-65 and described in Table 17-45.

Figure 17-65. ADMA Error Status Register (SD_ADMAES)

31	3	2	1	0
Reserved		LME	AES	
R-0		W-0	R/W-0	

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

Table 17-45. ADMA Error Status Register (SD_ADMAES) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Reserved bit field. Do not write any value.
2	LME		ADMA Length Mismatch Error: • While Block Count Enable is being set, the total data length specified by the Descriptor table is different from that specified by the Block Count and Block Length. • Total data length cannot be divided by the block length.
		0	No error
		1	Error
1-0	AES		ADMA Error State. This field indicates the state of ADMA when an error occurred during an ADMA data transfer. This field never indicates "10" because ADMA never stops in this state.
		Read 0	ST_STOP (Stop DMA). Contents of the SYS_SDR register
		Write 1h	ST_STOP (Stop DMA). Points to the error descriptor.
		Read 2h	Never set this state. (Not used)
		Write 3h	ST_TFR (Transfer Data). Points to the 'next' of the error descriptor.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.5.30 ADMA System Address Low Bits Register (SD_ADMASAL)

This register holds the byte address of the executing command of the Descriptor table. The 32-bit Address Descriptor uses the lower 32 bits of this register. At the start of ADMA, the Host Driver shall set the start address of the Descriptor table.

The ADMA System address low bits register (SD_ADMASAL) is shown in Figure 17-66 and described in Table 17-46.

Figure 17-66. ADMA System Address Low Bits (SD_ADMASAL)

31	0
ADMA_A32B	
R/W-0	

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

Table 17-46. ADMA System Address Low Bits (SD_ADMASAL) Field Descriptions

Bit	Field	Value	Description
31-0	ADMA_A32B		The ADMA increments this register address, which points to the next line, whenever fetching a Descriptor line. When the ADMA Error Interrupt is generated, this register holds the valid Descriptor address depending on the ADMA state. The Host Driver shall program the Descriptor Table on a 32-bit boundary and set the 32-bit boundary address to this register. ADMA2 ignores the lower 2 bits of this register and assumes it to be 00b.

17.5.31 ADMA System Address High Bits Register (SD_ADMASAH)

The ADMA System address high bits register (SD_ADMASAH) is shown in Figure 17-66 and described in Table 17-46.

Figure 17-67. ADMA System Address High Bits Register (SD_ADMASAH)

31	0
ADMA_A32B	
R/W-0	

LEGENDR/W = Read/Write; R = Read only; -*n* = value after reset

Table 17-47. ADMA System Address High Bits Register (SD_ADMASAH) Field Descriptions

Bit	Field	Value	Description
31-0	ADMA_A32B		

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

17.5.32 Versions Register (SD_REV)

This register contains the hard coded RTL vendor revision number, the version number of SD specification compliancy and a slot status bit.

- SD_REV[31:16] = Host Controller Version
- SD_REV[15:0] = Slot Interrupt Status

The versions register (SD_REV) is shown in Figure 17-68 and described in Table 17-48.

Figure 17-68. Versions Register (SD_REV)

31–24	23–16	15–1	0
VREV	SREV	Reserved	SIS
R-x	R-0	R-0	R-0

LEGENDR/W = Read/Write; R = Read only; *-n* = value after reset

Table 17-48. Versions Register (SD_REV) Field Descriptions

Bit	Field	Value	Description
31-24	VREV	0	Vendor Version Number. Bits [7:4] is the major revision, bits [3:0] is the minor revision. Examples: 10h for 1.0 21h for 2.1
23-16	SREV	Read 0	Specification Version Number. This status indicates the Standard SD Host Controller Specification Version. The upper and lower 4-bits indicate the version. SD Host Specification Version 1.0
15-1	Reserved	0	Reserved bit field. Do not write any value.
0	SIS		Slot Interrupt Status. This status bit indicates the inverted state of interrupt signal for the module. By a power on reset or by setting a software reset for all (SD_SYSCTL[24] SRA), the interrupt signal shall be deasserted and this status shall read 0.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Chapter 18

SPRUH73–July 2011

Universal Asynchronous Receiver/Transmitter (UART)

This chapter describes the UART of the device.

Topic **Page**

18.1 Introduction **2414**
18.2 Intregration **2416**
18.3 Functional Description **2419**
18.4 UART/IrDA/CIR Basic Programming Model **2453**
18.5 UART Registers **2462**

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.1 Introduction

18.1.1 UART Mode Features

The general features of the UART/IrDA module when operating in UART mode are:

- 16C750 compatibility
- Baud rate from 300 bps up to 3.6864 Mbps
- Auto-baud between 1200 bps and 115.2 Kbps
- Software/Hardware flow control
  - Programmable Xon/Xoff characters
  - Programmable Auto-RTS and Auto CTS
- Programmable serial interface characteristics
  - 5, 6, 7, or 8-bit characters
  - Even, odd, mark (always 1), space (always 0), or no parity (non-parity bit frame) bit generation and detection
  - 1, 1.5, or 2 stop bit generation
- False start bit detection
- Line break generation and detection
- Modem control functions (CTS, RTS, DSR, DTR, RI, and DCD)
- Fully prioritized interrupt system controls
- Internal test and loopback capabilities

18.1.2 IrDA Mode Features

The general features of the UART/IrDA when operating in IrDA or CIR mode are:

- Support of IrDA 1.4 slow infrared (SIR), medium infrared (MIR) and fast infrared (FIR) communications (very fast infrared (VFIR) is not supported).
- Frame formatting: addition of variable xBOF characters and EOF characters
- Uplink/downlink CRC generation/detection
- Asynchronous transparency (automatic insertion of break character)
- 8-entry status FIFO (with selectable trigger levels) available to monitor frame length and frame errors
- Framing error, cyclic redundancy check (CRC) error, illegal symbol (FIR), abort pattern (SIR, MIR) detection
- Support of consumer infrared (CIR) for remote control applications.
- Transmit and receive
- Free data format (supports any remote control private standards)
- Selectable bit rate
- Configurable carrier frequency
- 1/2, 5/12, 1/3 or 1/4 carrier duty cycle

18.1.3 Unsupported UART Features

The following UART/IrDA module features are not supported in this device.

Table 18-1. Unsupported UART Features

Feature	Reason
Full modem control on UART0	DCD, DSR, DTR, RI not pinned-out
Full modem control on UART2-5	DCD, DSR, DTR, RI not pinned-out
Device wake-up on UART1-5	Wake-up not supported - no SWake connection

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.2 Intregration

This device contains 6 instantiations of the UART/IrDA (UARTIRDAOCP) peripheral. There are six UART modules called UART0 – UART5. UART0 provides wakeup capability. Only UART 1 provides full modem control signals. All UARTs support IrDA and CIR modes and RTS/CTS flow control (subject to pin muxing configuration). Figure 18-1 shows an example of system connectivity using UART communication with hardware handshake.

Figure 18-1. UART/IrDA Module — UART Application

Figure 18-2 shows an example of system connectivity using infrared communication with remote control (consumer infrared).

Figure 18-2. UART/IrDA Module — IrDA/CIR Application

18.2.1 UART Connectivity Attributes

The general connectivity attributes for each of the UART modules are shown in Table 18-2 and Table 18-3.

Table 18-2. UART0 Connectivity Attributes

Attributes	Type
Power Domain	Wake-Up Domain
Clock Domain	PD_WKUP_L4_WKUP_GCLK (OCP) PD_WKUP_UART0_GFCLK (Func)
Reset Signals	WKUP_DOM_RST_N
Idle/Wakeup Signals	Smart Idle / Wakeup
Interrupt Requests	1 interrupt to MPU Subsystem, PRUSS and WakeM3
DMA Requests	2 DMA requests to EDMA
Physical Address	L4 Wakeup slave port

Table 18-3. UART1–5 Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_L4LS_GCLK (OCP) PD_PER_UART_GFCLK (Func)
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	UART1-2 1 interrupt to MPU Subsystem and PRUSS UART3-5 1 interrupt to MPU Subsystem only
DMA Requests	2 DMA requests to EDMA
Physical Address	L4 Peripheral slave port

18.2.2 UART Clock and Reset Management

The UART modules use separate functional and bus interface clocks.

Table 18-4. UART0 Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
CLK			
Interface clock	100 MHz	CORE_CLKOUTM4 / 2	pd_wkup_l4_wkup_gclk

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Table 18-4. UART0 Clock Signals (continued)

Clock Signal	Max Freq	Reference / Source	Comments
From PRCM			
FCLK			
Functional clock	48 MHz	PER_CLKOUTM2 / 4	pd_wkup_uart0_gfclk
from PRCM			

Table 18-5. UART1–5 Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
CLK Interface clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_l4ls_gclk From PRCM
FCLK Functional clock	48 MHz	PER_CLKOUTM2 / 4	pd_per_uart_gfclk From PRCM

For UART operation, the functional clock is used to produce a baud rate up to 3.6M bits/s. Table 18-6 lists the supported baud rates, the requested divider, and the corresponding error versus the standard baud rate.

Table 18-6. UART Mode Baud and Error Rates

Baud rate	Over sampling	Divisor	Error (%)
300	16	10000	0
600	16	5000	0
1200	16	2500	0
2400	16	1250	0
4800	16	625	0
9600	16	313	0.16
14400	16	208	0.16
19200	16	156	0.16
28800	16	104	0.16
38400	16	78	0.16
57600	16	52	0.16
115200	16	26	0.16
230400	16	13	0.16
460800	13	8	0.16
921600	13	4	0.16
1843200	13	2	0.16
3000000	16	1	0
3686400	13	1	0.16

For IrDA operation, the internal functional clock divisor allows generation of SIR, MIR, or FIR baud rates as shown in Table 18-7.

Table 18-7. IrDA Mode Baud and Error Rates

Baud rate	IR mode	Encoding	Divisor	Error (%)
2400	SIR	3/16	1250	0
9600	SIR	3/16	312	0.16
19200	SIR	3/16	156	0.16
38400	SIR	3/16	78	0.16
57600	SIR	3/16	52	0.16
115200	SIR	3/16	26	0.16

Copyright © 2011, Texas Instruments Incorporated

Table 18-7. IrDA Mode Baud and Error Rates (continued)

Baud rate	IR mode	Encoding	Divisor	Error (%)
576000	MIR	1/4	2	0
1152000	MIR	1/4	1	0
4000000	FIR	4PPM	1	0

18.2.3 UART Pin List

The UART interface pins are listed in Table 18-8. Pin functionality depends on the selected operating mode of the module.

Table 18-8. UART Pin List

Pin	Type	Description
UARTx_RXD	I	UART / IrDA / CIR Receive Data
UARTx_TXD / IRTX / RCTX	OZ	UART / IrDA / CIR Transmit Data
UARTx_RTSn / SD	OZ	UART Request to Send / IrDA Mode
UARTx_CTSn	I	UART Clear to Send
UARTx_DTRn[(1)]	OZ	UART Data Terminal Ready
UARTx_DSRn[(1)]	I	UART Data Set Ready
UARTx_DCDn[(1)]	I	UART Data Carrier Detect
UARTx_RIN[(1)]	1	UART Ring Indicator

(1) UART1 only

The UART module includes mode selection outputs to control the function of multiplexed signal based on the selected operating mode. The IRDA_SELECT signal indicates IrDA mode and the RC signal indicates CIR mode. The signal muxing based on these control outputs is shown in Table 18-9.

Table 18-9. UART Muxing Control

RC	IRDA_SELECT	UARTx_TXD / IRTX / RCTX Function	UARTx_RTSn / SD Function	UARTx_CTSn Function	Mode
0	0	TXD	RTSn	CTSn	UART
0	1	IRTX	SD	not used	IrDA
1	0	RCTX	SD	not used	CIR
1	1	n/a	n/a	n/a	Not Valid

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.3 Functional Description

18.3.1 Block Diagram

The UART/IrDA/CIR module can be divided into three main blocks:

- FIFO management
- Mode selection
- Protocol formatting

FIFO management is common to all functions and enables the transmission and reception of data from the host processor point of view.

There are two modes:

- Function mode: Routes the data to the chosen function (UART, IrDA, or CIR) and enables the mechanism corresponding to the chosen function
- Register mode: Enables conditional access to registers

For more information about mode configuration, see Section 18.3.7, *Mode Selection*.

Protocol formatting has three subcategories:

- Clock generation: The 48-MHz input clock generates all necessary clocks.
- Data formatting: Each function uses its own state-machine that is responsible for the transition between FIFO data and frame data associated with it.
- Interrupt management: Different interrupt types are generated depending on the chosen function:
  - UART mode interrupts: Seven interrupts prioritized in six different levels
  - IrDA mode interrupts: Eight interrupts. The interrupt line is activated when any interrupt is generated (there is no priority).
  - CIR mode interrupts: A subset of existing IrDA mode interrupts is used.

  In each mode, when an interrupt is generated, the UART_IIR register indicates the interrupt type.

In parallel with these functional blocks, a power-saving strategy exists for each function.

Figure 18-3 is the UART/IrDA/CIR block diagram.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 18-3. UART/IrDA/CIR Functional Specification Block Diagram

18.3.2 Clock Configuration

Each UART uses a 48-MHz functional clock for its logic and to generate external interface signals. Each UART uses an interface clock for register accesses. The PRCM module generates and controls all these clocks (for more information, see , *Clock Domain Module Attributes*, in Chapter 7, *Power, Reset, and Clock Management*).

The idle and wake-up processes use a handshake protocol between the PRCM and the UART (for a description of the protocol, see , *Module-Level Clock Managment*, in Chapter 7, *Power, Reset, and Clock Management*). The UARTi.UART_SYSC[4:3] IDLEMODE bit field controls UART idle mode.

18.3.3 Software Reset

The UARTi.UART_SYSC[1] SOFTRESET bit controls the software reset; setting this bit to 1 triggers a software reset functionally equivalent to hardware reset.

18.3.4 Power Management

18.3.4.1 UART Mode Power Management

18.3.4.1.1 Module Power Saving

In UART modes, sleep mode is enabled by setting the UARTi.UART_IER[4] SLEEP_MODE bit to 1 (when the UARTi.UART_EFR[4] ENHANCED_EN bit is set to 1).

Sleep mode is entered when all the following conditions exist:

- The serial data input line, uarti_rx, is idle.
- The TX FIFO and TX shift register are empty.
- The RX FIFO is empty.
- The only pending interrupts are THR interrupts.

Copyright © 2011, Texas Instruments Incorporated

Sleep mode is a good way to lower UART power consumption, but this state can be achieved only when the UART is set to modem mode. Therefore, even if the UART has no key role functionally, it must be initialized in a functional mode to take advantage of sleep mode.

In sleep mode, the module clock and baud rate clock are stopped internally. Because most registers are clocked by these clocks, this greatly reduces power consumption. The module wakes up when a change is detected on the uarti_rx line, when data is written to the TX FIFO, and when there is a change in the state of the modem input pins.

An interrupt can be generated on a wake-up event by setting the UARTi.UART_SCR[4] RX_CTS_WU_EN bit to 1. To understand how to manage the interrupt, see Section 18.3.5.1.2, *Wake-Up Interrupt*.

NOTE: There must be no writing to the divisor latches, UARTi.UART_DLL and UARTi.UART_DLH, to set the baud clock (BCLK) while in sleep mode. It is advisable to disable sleep mode using the UARTi.UART_IER[4] SLEEP_MODE bit before writing to the UARTi.UART_DLL register or the UARTi.UART_DLH register.

18.3.4.1.2 System Power Saving

Sleep and auto-idle modes are embedded power-saving features. Power-reduction techniques can be applied at the system level by shutting down certain internal clock and power domains of the device.

The UART supports an idle req/idle ack handshaking protocol used at the system level to shut down the UART clocks in a clean and controlled manner and to switch the UART from interrupt-generation mode to wake-up generation mode for unmasked events (see the UARTi.UART_SYSC[2] ENAWAKEUP bit and the UARTi.UART_WER register).

For more information, see , *Module Level Clock Managment*, in Chapter 7, *Power, Reset, and Clock Management*.

18.3.4.2 IrDA Mode Power Management (UART3 Only)

18.3.4.2.1 Module Power Saving

In IrDA modes, sleep mode is enabled by setting the UART3.MDR[3] IR_SLEEP bit to 1.

Sleep mode is entered when all the following conditions exist:

- The serial data input line, uart3.rx_irrx, is idle.
- The TX FIFO and TX shift register are empty.
- The RX FIFO is empty.
- No interrupts are pending except THR interrupts.

The module wakes up when a change is detected on the uart3_rx_irrx line or when data is written to the TX FIFO.

18.3.4.2.2 System Power Saving

System power saving for the IrDA mode has the same function as for the UART mode (see Section 18.3.4.1.2, *System Power Saving*).

18.3.4.3 CIR Mode Power Management (UART3 Only)

18.3.4.3.1 Module Power Saving

Module power saving for the CIR mode has the same function as for the IrDA mode (see Section 18.3.4.2.1, *Module Power Saving*).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.3.4.3.2 System Power Saving

System power saving for the CIR mode has the same function as for the UART mode (see Section 18.3.4.1.2, *System Power Saving*).

18.3.4.4 Local Power Management

Table 18-10 describes power-management features available for the UART.

NOTE: For information about source clock gating and sleep/wake-up transitions description, see , *Module-Level Clock Managment*, in Chapter 7, *Power, Reset, and Clock Management*.

Table 18-10. Local Power-Management Features

Feature	Registers	Description
Clock autogating	UART_SYSC[0] AUTOIDLE	This bit allows local power optimization in the module by gating the UARTi_ICLK clock on interface activity or gating the UARTi_FCLK clock on internal activity.
Slave idle modes	UART_SYSC[4:3] IDLEMODE	Force-idle, no-idle, smart-idle, and smart-idle wakeup-capable modes are available
Clock activity	N/A	Feature not available
Master standby modes	N/A	Feature not available
Global wake-up enable	UART_SYSC[2] ENAWAKEUP	This bit enables the wake-up feature at module level.
Wake-Up sources enable	N/A	Feature not available

18.3.5 Interrupt Requests

18.3.5.1 UART Mode Interrupt Management

18.3.5.1.1 UART Interrupts

UART mode includes seven possible interrupts prioritized to six levels.

When an interrupt is generated, the interrupt identification register (UARTi.UART_IIR) sets the UARTi.UART_IIR[0] IT_PENDING bit to 0 to indicate that an interrupt is pending, and indicates the type of interrupt through the UARTi.UART_IIR[5:1] bit field. Table 18-11 summarizes the interrupt control functions.

Table 18-11. UART Mode Interrupts

UART_IIR[5:0]	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Method
000001	None	None	None	None
000110	1	Receiver line status	OE, FE, PE, or BI errors occur in characters in the RX FIFO.	FE, PE, BI: Read the UART_RHR register. OE: Read the UART_LSR register.
001100	2	RX time-out	Stale data in RX FIFO	Read the UART_RHR register.
000100	2	RHR interrupt	DRDY (data ready) (FIFO disable) RX FIFO above trigger level (FIFO enable)	Read the UART_RHR register until the interrupt condition disappears.
000010	3	THR interrupt	TFE (UART_THR empty) (FIFO disable) TX FIFO below trigger level (FIFO enable)	Write to the UART_THR until the interrupt condition disappears.
000000	4	Modem status	See the UART_MSR register.	Read the UART_MSR register.

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Table 18-11. UART Mode Interrupts (continued)

UART_IIR[5:0]	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Method
010000	5	XOFF interrupt/special character interrupt	Receive XOFF characters/special character	Receive XON character(s), if XOFF interrupt/read of the UART_IIR register, if special character interrupt.
100000	6	CTS, RTS	RTS pin or CTS pin change state from active (low) to inactive (high).	Read the UART_IIR register.

For the receiver-line status interrupt, the RX_FIFO_STS bit (UARTi.UART_LSR[7]) generates the interrupt.

For the XOFF interrupt, if an XOFF flow character detection caused the interrupt, the interrupt is cleared by an XON flow character detection. If special character detection caused the interrupt, the interrupt is cleared by a read of the UARTi.UART_IIR register.

18.3.5.1.2 Wake-Up Interrupt

Wake-up interrupt is a special interrupt that works differently from other interrupts. This interrupt is enabled when the UARTi.UART_SCR[4] RX_CTS_WU_EN bit is set to 1. The UARTi.UART_IIR register is not modified when this occurs; the UART3.UART_SSR[1] RX_CTS_WU_STS bit must be checked to detect a wake-up event.

When a wake-up interrupt occurs, it can be cleared only by resetting the UARTi.UART_SCR[4] RX_CTS_WU_EN bit. This bit must be reenabled (set to 1) after the current wake-up interrupt event is processed to detect the next incoming wake-up event.

18.3.5.2 IrDA Mode Interrupt Management

18.3.5.2.1 IrDA Interrupts

The IrDA function generates interrupts. All interrupts can be enabled and disabled by writing to the appropriate bit in the interrupt enable register (UART3.UART_IER). The interrupt status of the device can be checked by reading the interrupt identification register (UART3.UART_IIR).

UART, IrDA, and CIR modes have different interrupts in the UART/IrDA/CIR module and, therefore, different UART3.UART_IER and UART3.UART_IIR mappings, depending on the selected mode.

IrDA modes have eight possible interrupts (see Table 18-12). The interrupt line is activated when any interrupt is generated (there is no priority).

Table 18-12. IrDA Mode Interrupts

UART_IIR Bit	Interrupt Type	Interrupt Source	Interrupt Reset Method
0	RHR interrupt	DRDY (data ready) (FIFO disable) RX FIFO above trigger level (FIFO enable)	Read the UART_RHR register until the interrupt condition disappears.
1	THR interrupt	TFE (UART_THR empty) (FIFO disable) TX FIFO below trigger level (FIFO enable)	Write to the UART_THR until the interrupt condition disappears.
2	Last byte in RX FIFO	Last byte of frame in RX FIFO is available to be read at the RHR port.	Read the UART_RHR register.
3	RX overrun	Write to the UART_RHR register when the RX FIFO is full.	Read UART_RESUME register.
4	Status FIFO interrupt	Status FIFO triggers level reached.	Read STATUS FIFO.

Copyright © 2011, Texas Instruments Incorporated

Table 18-12. IrDA Mode Interrupts (continued)

UART_IIR Bit	Interrupt Type	Interrupt Source	Interrupt Reset Method
5	TX status	1. UART_THR empty before EOF sent. Last bit of transmission of the IrDA frame occurred, but with an underrun error. OR 2. Transmission of the last bit of the IrDA frame completed successfully.	1. Read the UART_RESUME register. OR 2. Read the UART_IIR register.
6	Receiver line status interrupt	CRC, ABORT, or frame-length error is written into the STATUS FIFO.	Read the STATUS FIFO (read until empty - maximum of eight reads required).
7	Received EOF	Received end-of-frame	Read the UART_IIR register.

18.3.5.2.2 Wake-Up Interrupts

The wake-up interrupt for IrDA mode has the same function as that for UART mode (see Section 18.3.5.1.2, *Wake-Up Interrupt*).

CAUTION

Wake-Up interface implementation in this mode is based on the UARTi_SIDLEACK low-to-high transition instead of the UARTi_SIDLEACK state.

This does not ensure wake-up event generation as expected when configured in smart-idle mode, and the system wakes up for a short period.

18.3.5.3 CIR Mode Interrupt Management

18.3.5.3.1 CIR Interrupts

The CIR function generates interrupts that can be enabled and disabled by writing to the appropriate bit in the interrupt enable register (UART3.UART_IER). The interrupt status of the device can be checked by reading the interrupt identification register (UART3.UART_IIR).

UART, IrDA, and CIR modes have different interrupts in the UART/IrDA/CIR module and, therefore, different UART3.UART_IER and UART3.UART_IIR mappings, depending on the selected mode.

Table 18-13 lists the interrupt modes to be maintained. In CIR mode, the sole purpose of the UART3.UART_IIR[5] bit is to indicate that the last bit of infrared data was passed to the uart3_cts_rctx pin.

Table 18-13. CIR Mode Interrupts

UART_IIR Bit Number	Interrupt Type	Interrupt Source	Interrupt Reset Method
0	N/A for CIR mode	N/A for CIR mode	N/A for CIR mode
1	THR interrupt	TFE (UART_THR empty) (FIFO disable) TX FIFO below trigger level (FIFO enable)	Write to the UART_THR register until the interrupt condition disappears.
2	N/A for CIR mode	N/A for CIR mode	N/A for CIR mode
3	N/A for CIR mode	N/A for CIR mode	N/A for CIR mode
4	N/A for CIR mode	N/A for CIR mode	N/A for CIR mode
5	TX status	Transmission of the last bit of the frame is complete successfully.	Read the UART_IIR register.

Copyright © 2011, Texas Instruments Incorporated

Table 18-13. CIR Mode Interrupts (continued)

UART_IIR Bit Number	Interrupt Type	Interrupt Source	Interrupt Reset Method
6	N/A for CIR mode	N/A for CIR mode	N/A for CIR mode
7	N/A for CIR mode	N/A for CIR mode	N/A for CIR mode

18.3.5.3.2 Wake-Up Interrupts

The wake-up interrupt for CIR mode has the same function as that for UART mode (see Section 18.3.5.1.2, *Wake-Up Interrupt*).

18.3.6 FIFO Management

The FIFO is accessed by reading and writing the UARTi.UART_RHR and UARTi.UART_THR registers. Parameters are controlled using the FIFO control register (UARTi.UART_FCR) and supplementary control register (UARTi.UART_SCR). Reading the UARTi.UART_SSR[0] TX_FIFO_FULL bit at 1 means the FIFO is full.

The UARTi.UART_TLR register controls the FIFO trigger level, which enables DMA and interrupt generation. After reset, transmit (TX) and receive (RX) FIFOs are disabled; thus, the trigger level is the default value of 1 byte. Figure 18-4 shows the FIFO management registers.

NOTE: Data in the UARTi.UART_RHR register is not overwritten when an overflow occurs.

NOTE: The UARTi.UART_SFLSR, UARTi.UART_SFREGL, and UARTi.UART_SFREGH status registers are used in IrDA mode only. For use, see Section 18.3.8.2.3, *IrDA Data Formatting*.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 18-4. FIFO Management Registers

DRAFT

Copyright © 2011, Texas Instruments Incorporated

18.3.6.1 FIFO Trigger

18.3.6.1.1 Transmit FIFO Trigger

Table 18-14 lists the TX FIFO trigger level settings.

Table 18-14. TX FIFO Trigger Level Setting Summary

UART_SCR[6]	UART_TLR[3:0]	TX FIFO Trigger Level
0	= 0x0	Defined by the UARTi.UART_FCR[5:4] TX_FIFO_TRIG bit field (8,16, 32, or 56 spaces)
0	!= 0x0	Defined by the UARTi.UART_TLR[3:0] TX_FIFO_TRIG_DMA bit field (from 4 to 60 spaces with a granularity of 4 spaces)
1	Value	Defined by the concatenated value of TX_FIFO_TRIG_DMA and TX_FIFO_TRIG (from 1 to 63 spaces with a granularity of 1 space) **Note:** The combination of TX_FIFO_TRIG_DMA = 0x0 and TX_FIFO_TRIG = 0x0 (all zeros) is not supported (minimum of one space required). All zeros result in unpredictable behavior.

18.3.6.1.2 Receive FIFO Trigger

Table 18-15 lists the RX FIFO trigger level settings.

Table 18-15. RX FIFO Trigger Level Setting Summary

UART_SCR[7]	UART_TLR[7:4]	RX FIFO Trigger Level
0	= 0x0	Defined by the UARTi.UART_FCR[7:6] RX_FIFO_TRIG bit field (8,16, 56, or 60 characters)
0	!= 0x0	Defined by the UARTi.UART_TLR[7:4] RX_FIFO_TRIG_DMA bit field (from 4 to 60 characters with a granularity of 4 characters)
1	Value	Defined by the concatenated value of RX_FIFO_TRIG_DMA and RX_FIFO_TRIG (from 1 to 63 characters with a granularity of 1 character) **Note:** The combination of RX_FIFO_TRIG_DMA = 0x0 and RX_FIFO_TRIG = 0x0 (all zeros) is not supported (minimum of one character required). All zeros result in unpredictable behavior.

The receive threshold is programmed using the UARTi.UART_TCR[7:4] RX_FIFO_TRIG_START and UARTi.UART_TCR[3:0] RX_FIFO_TRIG_HALT bit fields:

- Trigger levels from 0 to 60 bytes are available with a granularity of 4 (trigger level = 4 x [4-bit register value]).
- To ensure correct device operation, ensure that RX_FIFO_TRIG_HALT RX_FIFO_TRIG when auto-RTS is enabled.

  Delay = [4 + 16 x (1 + CHAR_LENGTH + Parity + Stop 0.5)] x Baud_rate + 4 x FCLK

  NOTE: The RTS signal is deasserted after the UART module receives the data over RX_FIFO_TRIG_HALT. Delay means how long the UART module takes to deassert the RTS signal after reaching RX_FIFO_TRIG_HALT.

- In FIFO interrupt mode with flow control, ensure that the trigger level to HALT transmission is greater than or equal to the RX FIFO trigger level (the UARTi.UART_TCR[7:4] RX_FIFO_TRIG_START bit field or the UARTi.UART_FCR[7:6] RX_FIFO_TRIG bit field); otherwise, FIFO operation stalls. In FIFO DMA mode with flow control, this concept does not exist, because a DMA request is sent when a byte is received.

18.3.6.2 FIFO Interrupt Mode

In FIFO interrupt mode (the FIFO control register UARTi.UART_FCR[0] FIFO_EN bit is set to 1 and

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

relevant interrupts are enabled by the UARTi.UART_IER register), an interrupt signal informs the processor of the status of the receiver and transmitter. These interrupts are raised when the RX/TX FIFO threshold (the UARTi.UART_TLR[7:4] RX_FIFO_TRIG_DMA and UARTi.UART_TLR[3:0] TX_FIFO_TRIG_DMA bit fields or the UARTi.UART_FCR[7:6] RX_FIFO_TRIG and UARTi.UART_FCR[5:4] TX_FIFO_TRIG bit fields, respectively) is reached.

The interrupt signals instruct the MPU to transfer data to the destination (from the UART in receive mode and/or from any source to the UART FIFO in transmit mode).

When UART flow control is enabled with interrupt capabilities, the UART flow control FIFO threshold (the UARTi.UART_TCR[3:0] RX_FIFO_TRIG_HALT bit field) must be greater than or equal to the RX FIFO threshold.

Figure 18-5 shows the generation of the RX FIFO interrupt request.

Figure 18-5. RX FIFO Interrupt Request Generation

In receive mode, no interrupt is generated until the RX FIFO reaches its threshold. Once low, the interrupt can be deasserted only when the MPU has handled enough bytes to put the FIFO level below threshold. The flow control threshold is set at a higher value than the FIFO threshold.

Figure 18-6 shows the generation of the TX FIFO interrupt request.

Figure 18-6. TX FIFO Interrupt Request Generation

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

In transmit mode, an interrupt request is automatically asserted when the TX FIFO is empty. This request is deasserted when the TX FIFO crosses the threshold level. The interrupt line is deasserted until a sufficient number of elements is transmitted to go below the TX FIFO threshold.

18.3.6.3 FIFO Polled Mode Operation

In FIFO polled mode (the UARTi.UART_FCR[0] FIFO_EN bit is set to 0 and the relevant interrupts are disabled by the UARTi.UART_IER register), the status of the receiver and transmitter can be checked by polling the line status register (UARTi.UART_LSR).

This mode is an alternative to the FIFO interrupt mode of operation in which the status of the receiver and transmitter is automatically determined by sending interrupts to the MPU.

18.3.6.4 FIFO DMA Mode Operation

Although DMA operation includes four modes (DMA modes 0 through 3), the information in assumes that mode 1 is used. (Mode 2 and mode 3 are legacy modes that use only one DMA request for each module.)

In mode 2, the remaining DMA request is used for RX. In mode 3, the remaining DMA request is used for TX.

DMA requests in mode 2 and mode 3 use the following signals:

- S_DMA_48
- S_DMA_50
- S_DMA_52/D_DMA_10
- S_DMA_54

The following signals are not used by the module in mode 2 and mode 3:

- S_DMA_49
- S_DMA_51
- S_DMA_53/D_DMA_11
- S_DMA_55

These signals can be selected as follows:

- When the UARTi.UART_SCR[0] DMA_MODE_CTL bit is set to 0, setting the UARTi.UART_FCR[3] DMA_MODE_CTL bit to 0 enables DMA mode 0. Setting the DMA_MODE_CTL bit to 1 enables DMA mode 1.
- When the DMA_MODE_CTL bit is set to 1, the UARTi.UART_FCR[2:1] DMA_MODE_2 bit field determines DMA mode 0 to mode 3 based on the supplementary control register (UART_SCR) description.

For example:

- If no DMA operation is desired, set the DMA_MODE_CTL bit to 1 and the DMA_MODE_2 bit field to 0x0. (The DMA_MODE bit is discarded.)
- If DMA mode 1 is desired, set the DMA_MODE_CTL bit to 0 and the DMA_MODE bit to 1, or set the DMA_MODE_CTL bit to 1 and the DMA_MODE_2 bit field to 01. (The DMA_MODE bit is discarded.)

If the FIFOs are disabled (the UARTi.UART_FCR[0] FIFO_EN bit is set to 0), the DMA occurs in single-character transfers.

When DMA mode 0 is programmed, the signals associated with DMA operation are not active.

Depending on UART_MDR3[2] SET_DMA_TX_THRESHOLD, the threshold can be programmed different ways:

- SET_TX_DMA_THRESHOLD = 1:

  The threshold value will be the value of the UART_TX_DMA_THRESHOLD register. If SET_TX_DMA_THRESHOLD + TX trigger spaces 64, then the default method of threshold is used: threshold value = TX FIFO size.
- SET_TX_DMA_THRESHOLD = 0:

Copyright © 2011, Texas Instruments Incorporated

The threshold value = TX FIFO size TX trigger space. The TX DMA line is asserted if the TX FIFO level is lower then the threshold. It remains asserted until TX trigger spaces number of bytes are written into the FIFO. The DMA line is then deasserted and the FIFO level is compared with the threshold value.

18.3.6.4.1 DMA Transfers (DMA Mode 1, 2, or 3)

Figure 18-7 through Figure 18-10 show the supported DMA operations.

Figure 18-7. Receive FIFO DMA Request Generation (32 Characters)

In receive mode, a DMA request is generated when the RX FIFO reaches its threshold level defined in the trigger level register (UARTi.UART_TLR). This request is deasserted when the number of bytes defined by the threshold level is read by the sDMA.

In transmit mode, a DMA request is automatically asserted when the TX FIFO is empty. This request is deasserted when the number of bytes defined by the number of spaces in the UARTi.UART_TLR register is written by the sDMA. If an insufficient number of characters is written, the DMA request stays active.

Figure 18-8. Transmit FIFO DMA Request Generation (56 Spaces)

The DMA request is again asserted if the FIFO can receive the number of bytes defined by the UARTi.UART_TLR register.

Copyright © 2011, Texas Instruments Incorporated

The threshold can be programmed in a number of ways. Figure 18-8 shows a DMA transfer operating with a space setting of 56 that can arise from using the auto settings in the UARTi.UART_FCR[5:4] TX_FIFO_TRIG bit field or the UARTi.UART_TLR[3:0] TX_FIFO_TRIG_DMA bit field concatenated with the TX_FIFO_TRIG bit field.

The setting of 56 spaces in the UART/IrDA/CIR module must correlate with the settings of the sDMA so that the buffer does not overflow (program the DMA request size of the LH controller to equal the number of spaces in the UART/IrDA/CIR module).

Figure 18-9 shows an example with eight spaces to show the buffer level crossing the space threshold. The LH DMA controller settings must correspond to those of the UART/IrDA/CIR module.

Figure 18-9. Transmit FIFO DMA Request Generation (8 Spaces)

The next example shows the setting of one space that uses the DMA for each transfer of one character to the transmit buffer (see Figure 18-10). The buffer is filled faster than the baud rate at which data is transmitted to the TX pin. Eventually, the buffer is completely full and the DMA operations stop transferring data to the transmit buffer.

On two occasions, the buffer holds the maximum amount of data words; shortly after this, the DMA is disabled to show the slower transmission of the data words to the TX pin. Eventually, the buffer is emptied at the rate specified by the baud rate settings of the UARTi.UART_DLL and UARTi.UART_DLH registers.

The DMA settings must correspond to the system LH DMA controller settings to ensure correct operation of this logic.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 18-10. Transmit FIFO DMA Request Generation (1 Space)

The final example shows the setting of eight spaces, but setting the TX DMA threshold directly by setting the UART_MDR3[1] NONDEFAULT_FREQ bit and the UART_TX_DMA_THRESHOLD register (see Figure 18-11). In the example, UART_TX_DMA_THRESHOLD[2:0] TX_DMA_THRESHOLD = 3 and the trigger level is 8. The buffer is filled at a faster rate than the baud rate transmits data to the TX pin. The buffer is filled with 8 bytes and the DMA operations stop transferring data to the transmit buffer. When the buffer is emptied to the threshold level by transmission, the DMA operation activates again to fill the buffer with 8 bytes.

Eventually, the buffer is emptied at the rate specified by the baud rate settings of the UART_DLL and UART_DLH registers.

If the selected threshold level plus the trigger level exceed the maximum buffer size, the original TX DMA threshold method is used to prevent TX overrun, regardless of the value of the UART_MDR3[1] NONDEFAULT_FREQ bit.

The DMA settings must correspond to the settings of the system local host DMA controller to ensure the correct operation of this logic.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 18-11. Transmit FIFO DMA Request Generation Using Direct TX DMA Threshold Programming. (Threshold = 3; Spaces = 8)

18.3.6.4.2 DMA Transmission

Figure 18-12 shows DMA transmission.

Figure 18-12. DMA Transmission

1. Data to be transmitted are put in the device memory reserved for UART/IrDA/CIR transmission by the DMA:
   (a) Until the TX FIFO trigger level is not reached, a DMA request is generated
   (b) An element (1 byte) is transferred from the SDRAM to the TX FIFO at each DMA request (DMA element synchronization).
2. Data in the TX FIFO are automatically transmitted.
3. The end of the transmission is signaled by the UARTi.UART_THR empty (TX FIFO empty).

NOTE: In IrDA mode, the transmission does not end immediately after the TX FIFO empties, at which point the last data byte, the CRC field, and the stop flag still must be transmitted; thus, the end of transmission occurs a few milliseconds after the UARTi.UART_THR register empties.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.3.6.4.3 DMA Reception

Figure 18-13 shows DMA reception.

Figure 18-13. DMA Reception

1. Enable the reception.
2. Received data are put in the RX FIFO.
3. Data are transferred from the RX FIFO to the device memory by the DMA:
   (a) At each received byte, the RX FIFO trigger level (one character) is reached and a DMA request is generated.
   (b) An element (1 byte) is transferred from the RX FIFO to the SDRAM at each DMA request (DMA element synchronization).
4. The end of the reception is signaled by the EOF interrupt.

18.3.7 Mode Selection

18.3.7.1 Register Access Modes

18.3.7.1.1 Operational Mode and Configuration Modes

Register access depends on the register access mode, although register access modes are not correlated to functional mode selection. Three different modes are available:

- Operational mode
- Configuration mode A
- Configuration mode B

Operational mode is the selected mode when the function is active; serial data transfer can be performed in this mode.

Configuration mode A and configuration mode B are used during module initialization steps. These modes enable access to configuration registers, which are hidden in the operational mode. The modes are used when the module is inactive (no serial data transfer processed) and only for initialization or reconfiguration of the module.

The value of the UARTi.UART_LCR register determines the register access mode (see Table 18-16).

Table 18-16. UART/IrDA/CIR Register Access Mode Programming (Using UART_LCR)

Mode	Condition
Configuration mode A	UART_LCR[7] = 0x1 and UART_LCR[7:0] != 0xBF
Configuration mode B	UART_LCR[7] = 0x1 and UART_LCR[7:0] = 0xBF
Operational mode	UART_LCR[7] = 0x0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.3.7.1.2 Register Access Submode

In each access register mode (operational mode or configuration mode A/B), some register accesses are conditional on the programming of a submode (MSR_SPR, TCR_TLR, and XOFF). These registers are identified in , *UART/IrDA/CIR Register Manual*.

Table 18-17 through Table 18-19 summarize the register access submodes.

Table 18-17. Subconfiguration Mode A Summary

Mode	Condition
MSR_SPR	(UART_EFR[4] = 0x0 or UART_MCR[6] = 0x0)
TCR_TLR	UART_EFR[4] = 0x1 and UART_MCR[6] = 0x1

Table 18-18. Subconfiguration Mode B Summary

Mode	Condition
TCR_TLR	UART_EFR[4] = 0x1 and UART_MCR[6] = 0x1
XOFF	(UART_EFR[4] = 0x0 or UART_MCR[6] = 0x0)

Table 18-19. Suboperational Mode Summary

Mode	Condition
MSR_SPR	UART_EFR[4] = 0x0 or UART_MCR[6] = 0x0
TCR_TLR	UART_EFR[4] = 0x1 and UART_MCR[6] = 0x1

18.3.7.1.3 Registers Available for the Register Access Modes

Table 18-20 lists the names of the register bits in each access register mode. Gray shading indicates that the register does not depend on the register access mode (available in all modes).

Table 18-20. UART/IrDA/CIR Register Access Mode Overview

Address Offset	Registers					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x000	UART_DLL	UART_DLL	UART_DLL	UART_DLL	UART_RHR	UART_THR
0x004	UART_DLH	UART_DLH	UART_DLH	UART_DLH	UART_IER	UART_IER
0x008	UART_IIR	UART_FCR	UART_EFR	UART_EFR	UART_IIR	UART_FCR
0x00C	UART_LCR	UART_LCR	UART_LCR	UART_LCR	UART_LCR	UART_LCR
0x010	UART_MCR	UART_MCR	UART_XON1_ADDR1	UART_XON1_ADDR1	UART_MCR	UART_MCR
0x014	UART_LSR	–	UART_XON2_ADDR2	UART_XON2_ADDR2	UART_LSR	–
0x018	UART_MSR (1)/UART_TCR (2)	UART_TCR (2)	UART_TCR (2)/UART_XOFF1 (3)	UART_TCR (2)/UART_XOFF1 (3)	UART_MSR (1)/UART_TCR (2)	UART_TCR (2)
0x01C	UART_SPR (1)/UART_TLR (2)	UART_SPR (1)/UART_TLR (2)	UART_TLR (2)/UART_XOFF2 (3)	UART_TLR (2)/UART_XOFF2 (3)	UART_SPR (1)/UART_TLR (2)	UART_SPR (1)/UART_TLR (2)
0x020	UART_MDR1	UART_MDR1	UART_MDR1	UART_MDR1	UART_MDR1	UART_MDR1
0x024	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2
0x028	UART_SFLSR	UART_TXFLL	UART_SFLSR	UART_TXFLL	UART_SFLSR	UART_TXFLL

(1) MSR_SPR mode is active (see Section 18.3.7.1.2, *Register Access Submode*)
(2) TCR_TLR mode is active (see Section 18.3.7.1.2, *Register Access Submode*)
(3) XOFF mode is active (see Section 18.3.7.1.2, *Register Access Submode*)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 18-20. UART/IrDA/CIR Register Access Mode Overview (continued)

Address Offset	Registers					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x02C	UART_RESUME	UART_TXFLH	UART_RESUME	UART_TXFLH	UART_RESUME	UART_TXFLH
0x030	UART_SFREGL	UART_RXFLL	UART_SFREGL	UART_RXFLL	UART_SFREGL	UART_RXFLL
0x034	UART_SFREGH	UART_RXFLH	UART_SFREGH	UART_RXFLH	UART_SFREGH	UART_RXFLH
0x038	UART_UASR	–	UART_UASR	–	UART_BLR	UART_BLR
0x03C	–	–	–	–	UART_ACREG	UART_ACREG
0x040	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR
0x044	UART_SSR	–	UART_SSR	–	UART_SSR	–
0x048	–	–	–	–	UART_EBLR	UART_EBLR
0x050	UART_MVR	–	UART_MVR	–	UART_MVR	–
0x054	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC
0x058	UART_SYSS	–	UART_SYSS	–	UART_SYSS	–
0x05C	UART_WER	UART_WER	UART_WER	UART_WER	UART_WER	UART_WER
0x060	UART_CFPS	UART_CFPS	UART_CFPS	UART_CFPS	UART_CFPS	UART_CFPS
0x064	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL
0x068	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL
0x06C	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2
0x070	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2
0x074	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL
0x080	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3
0x084	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD

18.3.7.2 UART/IrDA (SIR, MIR, FIR)/CIR Mode Selection

To select a mode, set the UARTi.UART_MDR1[2:0] MODE_SELECT bit field (see Table 18-21).

Table 18-21. UART Mode Selection

Value	Mode
0x0:	UART 16x mode
0x1:	SIR mode (UART3 only)
0x2:	UART 16x auto-baud
0x3:	UART 13x mode
0x4:	MIR mode (UART3 only)
0x5:	FIR mode (UART3 only)
0x6:	CIR mode (UART3 only)

DRAFT

Copyright © 2011, Texas Instruments Incorporated

MODE_SELECT is effective when the module is in operational mode (see Section 18.3.7.1, *Register Access Modes*).

18.3.7.2.1 Registers Available for the UART Function

Only the registers listed in Table 18-22 are used for the UART function.

Table 18-22. UART Mode Register Overview[(1)] [(2)]

Address Offset	Registers					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x000	UART_DLL	UART_DLL	UART_DLL	UART_DLL	UART_RHR	UART_THR
0x004	UART_DLH	UART_DLH	UART_DLH	UART_DLH	UART_IER(UART)	UART_IER(UART)
0x008	UART_IIR	UART_FCR	UART_EFR[4]	UART_EFR[4]	UART_IIR(UART)	UART_FCR(UART)
0x00C	UART_LCR	UART_LCR	UART_LCR	UART_LCR	UART_LCR	UART_LCR
0x010	UART_MCR	UART_MCR	UART_XON1_ADDR1	UART_XON1_ADDR1	UART_MCR	UART_MCR
0x014	UART_LSR(UART)	–	UART_XON2_ADDR2	UART_XON2_ADDR2	UART_LSR(UART)	–
0x018	UART_MSR/UART_TCR	UART_TCR	UART_XOFF1/UART_TCR	UART_XOFF1/UART_TCR	UART_MSR/UART_TCR	UART_TCR
0x01C	UART_TLR/UART_SPR	UART_TLR/UART_SPR	UART_TLR/UART_XOFF2	UART_TLR/UART_XOFF2	UART_TLR/UART_SPR	UART_TLR/UART_SPR
0x020	UART_MDR1	UART_MDR1[2:0]	UART_MDR1[2:0]	UART_MDR1[2:0]	UART_MDR1[2:0]	UART_MDR1[2:0]
0x024	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2
0x028	–	–	–	–	–	–
0x02C	–	–	–	–	–	–
0x030	–	–	–	–	–	–
0x034	–	–	–	–	–	–
0x038	UART_UASR	–	UART_UASR	–	–	–
0x03C	–	–	–	–	–	–
0x040	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR
0x044	UART_SSR	–	UART_SSR	–	UART_SSR	–
0x048	–	–	–	–	–	–
0x050	UART_MVR	–	UART_MVR	–	UART_MVR	–
0x054	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC
0x058	UART_SYSS	–	UART_SYSS	–	UART_SYSS	–
0x05C	UART_WER	UART_WER	UART_WER	UART_WER	UART_WER	UART_WER
0x060	–	–	–	–	–	–
0x064	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL
0x068	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL
0x06C	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2
0x070	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2
0x074	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL
0x080	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3

(1) REGISTER_NAME(UART) notation indicates that the register exists for other functions (IrDA or CIR), but fields have different meanings for other functions (described separately in , *UART/IrDA/CIR Register Manual*).

(2) REGISTER_NAME[m:n] notation indicates that only register bits numbered m to n apply to the UART function.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 18-22. UART Mode Register Overview[(1) (2)] (continued)

Address Offset	Registers					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x084	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD

18.3.7.2.2 Registers Available for the IrDA Function (UART3 Only)

Only the registers listed in Table 18-23 are used for the IrDA function.

Table 18-23. IrDA Mode Register Overview[(1) (2)]

Address Offset	Registers					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x000	UART_DLL	UART_DLL	UART_DLL	UART_DLL	UART_RHR	UART_THR
0x004	UART_DLH	UART_DLH	UART_DLH	UART_DLH	UART_IER(IrDA)	UART_IER(IrDA)
0x008	UART_IIR	UART_FCR	UART_EFR[4]	UART_EFR[4]	UART_IIR(IrDA)	UART_FCR(IrDA)
0x00C	UART_LCR[7]	UART_LCR[7]	UART_LCR[7]	UART_LCR[7]	UART_LCR[7]	UART_LCR[7]
0x010	–	–	UART_XON1_ADDR1	UART_XON1_ADDR1	–	–
0x014	UART_LSR(IrDA)	–	UART_XON2_ADDR2	UART_XON2_ADDR2	UART_LSR(IrDA)	–
0x018	UART_MSR/UART_TCR	UART_TCR	UART_TCR	UART_TCR	UART_MSR/UART_TCR	UART_TCR
0x01C	UART_TLR/UART_SPR	UART_TLR/UART_SPR	UART_TLR	UART_TLR	UART_TLR/UART_SPR	UART_TLR/UART_SPR
0x020	UART_MDR1	UART_MDR1	UART_MDR1	UART_MDR1	UART_MDR1	UART_MDR1
0x024	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2
0x028	UART_SFLSR	UART_TXFLL	UART_SFLSR	UART_TXFLL	UART_SFLSR	UART_TXFLL
0x02C	UART_RESUME	UART_TXFLH	UART_RESUME	UART_TXFLH	UART_RESUME	UART_TXFLH
0x030	UART_SFREGL	UART_RXFLL	UART_SFREGL	UART_RXFLL	UART_SFREGL	UART_RXFLL
0x034	UART_SFREGH	UART_RXFLH	UART_SFREGH	UART_RXFLH	UART_SFREGH	UART_RXFLH
0x038	–	–	–	–	UART_BLR	UART_BLR
0x03C	–	–	–	–	UART_ACREG	UART_ACREG
0x040	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR
0x044	UART_SSR	–	UART_SSR	–	UART_SSR	–
0x048	–	–	–	–	UART_EBLR	UART_EBLR
0x050	UART_MVR	–	UART_MVR	–	UART_MVR	–
0x054	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC
0x058	UART_SYSS	–	UART_SYSS	–	UART_SYSS	–
0x05C	UART_WER[6:4]	UART_WER[6:4]	UART_WER[6:4]	UART_WER[6:4]	UART_WER[6:4]	UART_WER[6:4]
0x060	–	–	–	–	–	–

(1) REGISTER_NAME(UART) notation indicates that the register exists for other functions (IrDA or CIR), but fields have different meanings for other functions (described separately in , *UART/IrDA/CIR Register Manual*).

(2) REGISTER_NAME[m:n] notation indicates that only register bits numbered m to n apply to the UART function.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 18-23. IrDA Mode Register Overview[(1) (2)] (continued)

Address Offset	Registers					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x064	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL
0x068	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL
0x06C	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2
0x070	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2
0x074	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL
0x080	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3
0x084	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD

18.3.7.2.3 Registers Available for the CIR Function (UART3 Only)

Only the registers listed in Table 18-24 are used for the CIR function.

Table 18-24. CIR Mode Register Overview[(1) (2)]

Address Offset	Registers					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x000	UART_DLL	UART_DLL	UART_DLL	UART_DLL	–	UART_THR
0x004	UART_DLH	UART_DLH	UART_DLH	UART_DLH	UART_IER(CIR)	UART_IER(CIR)
0x008	UART_IIR	UART_FCR	UART_EFR	UART_EFR	UART_IIR(CIR)	UART_FCR(CIR)
0x00C	UART_LCR	UART_LCR[7]	UART_LCR[7]	UART_LCR[7]	UART_LCR[7]	UART_LCR[7]
0x010	–	–	–	–	–	–
0x014	UART_LSR(IrDA)	–	–	–	UART_LSR(IrDA)	–
0x018	UART_MSR/UART_TCR	UART_TCR	UART_TCR	UART_TCR	UART_MSR/UART_TCR	UART_TCR
0x01C	UART_TLR/UART_SPR	UART_TLR/UART_SPR	UART_TLR	UART_TLR	UART_TLR/UART_SPR	UART_TLR/UART_SPR
0x020	UART_MDR1[3:0]	UART_MDR1[3:0]	UART_MDR1[3:0]	UART_MDR1[3:0]	UART_MDR1[3:0]	UART_MDR1[3:0]
0x024	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2	UART_MDR2
0x028	–	–	–	–	–	–
0x02C	UART_RESUME	–	UART_RESUME	–	UART_RESUME	–
0x030	–	–	–	–	–	–
0x034	–	–	–	–	–	–
0x038	–	–	–	–	–	–
0x03C	–	–	–	–	UART_ACREG	UART_ACREG
0x040	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR
0x044	UART_SSR	–	UART_SSR	–	UART_SSR	–

(1) REGISTER_NAME(UART) notation indicates that the register exists for other functions (IrDA or CIR), but fields have different meanings for other functions (described separately in , *UART/IrDA/CIR Register Manual*).

(2) REGISTER_NAME[m:n] notation indicates that only register bits numbered m to n apply to the UART function.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 18-24. CIR Mode Register Overview[(1) (2)] (continued)

Address Offset	Registers					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x048	–	–	–	–	UART_EBLR	UART_EBLR
0x050	UART_MVR	–	UART_MVR	–	UART_MVR	–
0x054	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC
0x058	UART_SYSS	–	UART_SYSS	–	UART_SYSS	–
0x05C	UART_WER[6:4]	UART_WER[6:4]	UART_WER[6:4]	UART_WER[6:4]	UART_WER[6:4]	UART_WER[6:4]
0x060	UART_CFPS	UART_CFPS	UART_CFPS	UART_CFPS	UART_CFPS	UART_CFPS
0x064	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL
0x068	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL
0x06C	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2
0x070	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2
0x074	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL
0x080	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3
0x084	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD

18.3.8 Protocol Formatting

18.3.8.1 UART Mode

18.3.8.1.1 UART Clock Generation: Baud Rate Generation

The UART function contains a programmable baud generator and a set of fixed dividers that divide the 48-MHz clock input down to the expected baud rate.

Figure 18-14 shows the baud rate generator and associated controls.

Figure 18-14. Baud Rate Generation

Copyright © 2011, Texas Instruments Incorporated

CAUTION

Before initializing or modifying clock parameter controls (UARTi.UART_DLH, UARTi.UART_DLL), MODE_SELECT = DISABLE (UARTi.UART_MDR1[2:0]) must be set to 0x7. Failure to observe this rule can result in unpredictable module behavior.

18.3.8.1.2 Choosing the Appropriate Divisor Value

Two divisor values are:

- UART 16x mode: Divisor value = Operating frequency/(16x baud rate)
- UART 13x mode: Divisor value = Operating frequency/(13x baud rate)

Table 18-25 describes the UART baud rate settings.

Table 18-25. UART Baud Rate Settings (48-MHz Clock)

Baud Rate	Baud Multiple	DLH,DLL (Decimal)	DLH,DLL (Hex)	Actual Baud Rate	Error (%)
0.3 kbps	16x	10000	0x27, 0x10	0.3 kbps	0
0.6 kbps	16x	5000	0x13, 0x88	0.6 kbps	0
1.2 kbps	16x	2500	0x09, 0xC4	1.2 kbps	0
2.4 kbps	16x	1250	0x04, 0xE2	2.4 kbps	0
4.8 kbps	16x	625	0x02, 0x71	4.8 kbps	0
9.6 kbps	16x	312	0x01, 0x38	9.6153 kbps	+0.16
14.4 kbps	16x	208	0x00, 0xD0	14.423 kbps	+0.16
19.2 kbps	16x	156	0x00, 0x9C	19.231 kbps	+0.16
28.8 kbps	16x	104	0x00, 0x68	28.846 kbps	+0.16
38.4 kbps	16x	78	0x00, 0x4E	38.462 kbps	+0.16
57.6 kbps	16x	52	0x00, 0x34	57.692 kbps	+0.16
115.2 kbps	16x	26	0x00, 0x1A	115.38 kbps	+0.16
230.4 kbps	16x	13	0x00, 0x0D	230.77 kbps	+0.16
460.8 kbps	13x	8	0x00, 0x08	461.54 kbps	+0.16
921.6 kbps	13x	4	0x00, 0x04	923.08 kbps	+0.16
1.843 Mbps	13x	2	0x00, 0x02	1.846 Mbps	+0.16
3.6884 Mbps	13x	1	0x00, 0x01	3.6923 Mbps	+0.16

18.3.8.1.3 UART Data Formatting

The UART can use hardware flow control to manage transmission and reception. Hardware flow control significantly reduces software overhead and increases system efficiency by automatically controlling serial data flow using the RTS output and CTS input signals.

The UART is enhanced with the autobauding function. In control mode, autobauding lets the speed, the number of bits per character, and the parity selected be set automatically.

18.3.8.1.3.1 Frame Formatting

When autobauding is not used, frame format attributes must be defined in the UARTi.UART_LCR register.

Character length is specified using the UARTi.UART_LCR[1:0] CHAR_LENGTH bit field.

The number of stop-bits is specified using the UARTi.UART_LCR[2] NB_STOP bit.

The parity bit is programmed using the UARTi.UART_LCR[5:3] PARITY_EN, PARITY_TYPE_1, and PARITY_TYPE_2 bit fields (see Table 18-26).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 18-26. UART Parity Bit Encoding

PARITY_EN	PARITY_TYPE_1	PARITY_TYPE_2	Parity
0	N/A	N/A	No parity
1	0	0	Odd parity
1	1	0	Even parity
1	0	1	Forced 1
1	1	1	Forced 0

18.3.8.1.3.2 Hardware Flow Control

Hardware flow control is composed of auto-CTS and auto-RTS. Auto-CTS and auto-RTS can be enabled and disabled independently by programming the UARTi.UART_EFR[7:6] AUTO_CTS_EN and AUTO_RTS_EN bit fields, respectively.

With auto-CTS, uarti_cts must be active before the module can transmit data.

Auto-RTS activates the uarti_rts output only when there is enough room in the RX FIFO to receive data. It deactivates the uarti_rts output when the RX FIFO is sufficiently full. The HALT and RESTORE trigger levels in the UARTi.UART_TCR register determine the levels at which uarti_rts is activated and deactivated.

If auto-CTS and auto-RTS are enabled, data transmission does not occur unless the RX FIFO has empty space. Thus, overrun errors are eliminated during hardware flow control. If auto-CTS and auto-RTS are not enabled, overrun errors occur if the transmit data rate exceeds the RX FIFO latency.

- Auto-RTS:

  Auto-RTS data flow control originates in the receiver block. The RX FIFO trigger levels used in auto-RTS are stored in the UARTi.UART_TCR register. uarti_rts is active if the RX FIFO level is below the HALT trigger level in the UARTi.UART_TCR[3:0] RX_FIFO_TRIG_HALT bit field. When the RX FIFO HALT trigger level is reached, uarti_rts is deasserted. The sending device (for example, another UART) can send an additional byte after the trigger level is reached because it may not recognize the deassertion of RTS until it begins sending the additional byte.

  uarti_rts is automatically reasserted when the RX FIFO reaches the RESUME trigger level programmed by the UARTi.UART_TCR[7:4] RX_FIFO_TRIG_START bit field. This reassertion requests the sending device to resume transmission.

  In this case, uarti_rts is an active-low signal.

- Auto-CTS:

  The transmitter circuitry checks uarti_cts before sending the next data byte. When uarti_cts is active, the transmitter sends the next byte. To stop the transmitter from sending the next byte, uarti_cts must be deasserted before the middle of the last stop-bit currently sent.

  The auto-CTS function reduces interrupts to the host system. When auto-CTS flow control is enabled, the uarti_cts state changes do not have to trigger host interrupts because the device automatically controls its own transmitter. Without auto-CTS, the transmitter sends any data present in the transmit FIFO, and a receiver overrun error can result.

  In this case, uarti_cts is an active-low signal.

18.3.8.1.3.3 Software Flow Control

Software flow control is enabled through the enhanced feature register (UARTi.UART_EFR) and the modem control register (UARTi.UART_MCR). Different combinations of software flow control can be enabled by setting different combinations of the UARTi.UART_EFR[3:0] bit field (see Table 18-27).

Two other enhanced features relate to software flow control:

- XON any function (UARTi.UART_MCR[5]): Operation resumes after receiving any character after the XOFF character is recognized. If special character detect is enabled and special character is received after XOFF1, it does not resume transmission. The special character is stored in the RX FIFO.

Copyright © 2011, Texas Instruments Incorporated

NOTE: The XON-any character is written into the RX FIFO even if it is a software flow character.

- Special character (UARTi.UART_EFR[5]): Incoming data is compared to XOFF2. When the special character is detected, the XOFF interrupt (UARTi.UART_IIR[4]) is set, but it does not halt transmission. The XOFF interrupt is cleared by a read of UARTi.UART_IIR. The special character is transferred to the RX FIFO. Special character does not work with XON2, XOFF2, or sequential XOFFs.

Table 18-27. UART_EFR[3:0] Software Flow Control Options

Bit 3	Bit 2	Bit 1	Bit 0	TX, RX Software Flow Controls
0	0	X	X	No transmit flow control
1	0	X	X	Transmit XON1, XOFF1
0	1	X	X	Transmit XON2, XOFF2
1	1	X	X	Transmit XON1, XON2: XOFF1, XOFF2[(1)]
X	X	0	0	No receive flow control
X	X	1	0	Receiver compares XON1, XOFF1
X	X	0	1	Receiver compares XON2, XOFF2
X	X	1	1	Receiver compares XON1, XON2: XOFF1, XOFF2[(1)]

(1) In these cases, the XON1 and XON2 characters or the XOFF1 and XOFF2 characters must be transmitted/received sequentially with XON1/XOFF1 followed by XON2/XOFF2.
XON1 is defined in the UARTi.UART_XON1_ADDR1[7:0] XON_WORD1 bit field. XON2 is defined in the UARTi.UART_XON2_ADDR2[7:0] XON_WORD2 bit field.
XOFF1 is defined in the UARTi.UART_XOFF1[7:0] XOFF_WORD1 bit field. XOFF2 is defined in the UARTi.UART_XOFF2[7:0] XOFF_WORD2 bit field.

18.3.8.1.3.3.1 Receive (RX)

When software flow control operation is enabled, the UART compares incoming data with XOFF1/2 programmed characters (in certain cases, XOFF1 and XOFF2 must be received sequentially). When the correct XOFF characters are received, transmission stops after transmission of the current character completes. Detection of XOFF also sets the UARTi.UART_IIR[4] bit (if enabled by UARTi.UART_IER[5]) and causes the interrupt line to go low.

To resume transmission, an XON1/2 character must be received (in certain cases, XON1 and XON2 must be received sequentially). When the correct XON characters are received, the UARTi.UART_IIR[4] bit is cleared and the XOFF interrupt disappears.

NOTE: When a parity, framing, or break error occurs while receiving a software flow control character, this character is treated as normal data and is written to the RX FIFO.

When XON-any and special character detect are disabled and software flow control is enabled, no valid XON or XOFF characters are written to the RX FIFO. For example, when UARTi.UART_EFR[1:0] = 0x2, if XON1 and XOFF1 characters are received, they are not written to the RX FIFO.

When pairs of software flow characters are programmed to be received sequentially (UARTi.UART_EFR[1:0] = 0x3), the software flow characters are not written to the RX FIFO if they are received sequentially. However, received XON1/XOFF1 characters must be written to the RX FIFO if the subsequent character is not XON2/XOFF2.

18.3.8.1.3.3.2 Transmit (TX)

Two XOFF1 characters are transmitted when the RX FIFO passes the trigger level programmed by UARTi.UART_TCR[3:0]. As soon as the RX FIFO reaches the trigger level programmed by UARTi.UART_TCR[7:4], two XON1 characters are sent, so the data transfer recovers.

NOTE: If software flow control is disabled after an XOFF character is sent, the module transmits XON characters automatically to enable normal transmission.

Copyright © 2011, Texas Instruments Incorporated

The transmission of XOFF(s)/XON(s) follows the same protocol as transmission of an ordinary byte from the TX FIFO. This means that even if the word length is 5, 6, or 7 characters, the 5, 6, or 7 LSBs of XOFF1/2 and XON1/2 are transmitted. The 5, 6, or 7 bits of a character are seldom transmitted, but this function is included to maintain compatibility with earlier designs.

It is assumed that software flow control and hardware flow control are never enabled simultaneously.

18.3.8.1.3.4 Autobauding Modes

In autobauding mode, the UART can extract transfer characteristics (speed, length, and parity) from an "at" (AT) command (ASCII code). These characteristics are used to receive data after an AT and to send data.

The following AT commands are valid:

AT	DATA	CR
at	DATA	CR
A/		
a/		

A line break during the acquisition of the sequence AT is not recognized, and an echo function is not implemented in hardware.

A/ and a/ are not used to extract characteristics, but they must be recognized because of their special meaning. A/ or a/ is used to instruct the software to repeat the last received AT command; therefore, an a/ always follows an AT, and transfer characteristics are not expected to change between an AT and an a/.

When a valid AT is received, AT and all subsequent data, including the final CR (0x0D), are saved to the RX FIFO. The autobaud state-machine waits for the next valid AT command. If an a/ (A/) is received, the a/ (A/) is saved in the RX FIFO and the state-machine waits for the next valid AT command.

On the first successful detection of the baud rate, the UART activates an interrupt to signify that the AT (upper or lower case) sequence is detected. The UARTi.UART_UASR register reflects the correct settings for the baud rate detected. Interrupt activity can continue in this fashion when a subsequent character is received. Therefore, it is recommended that the software enable the RHR interrupt when using the autobaud mode.

The following settings are detected in autobaud mode with a module clock of 48 MHz:

- Speed:
  - 115.2K baud
  - 57.6K baud
  - 38.4K baud
  - 28.8K baud
  - 19.2K baud
  - 14.4K baud
  - 9.6K baud
  - 4.8K baud
  - 2.4K baud
  - 1.2K baud
- Length: 7 or 8 bits
- Parity: Odd, even, or space

NOTE: The combination of 7-bit character plus space parity is not supported.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Autobauding mode is selected when the UARTi.UART_MDR1[2:0] MODE_SELECT bit field is set to 0x2. In UART autobauding mode, UARTi.UART_DLL, UARTi.UART_DLH, and UARTi.UART_LCR[5:0] bit field settings are not used; instead, UART_UASR is updated with the configuration detected by the autobauding logic.

UART_UASR Autobauding Status Register Use

This register is used to set up transmission according to the characteristics of the previous reception instead of the UARTi.UART_LCR, UARTi.UART_DLL, and UARTi.UART_DLH registers when the UART is in autobauding mode.

To reset the autobauding hardware (to start a new AT detection) or to set the UART in standard mode (no autobaud), the UARTi.UART_MDR1[2:0] MODE_ SELECT bit field must be set to reset state (0x7) and then to the UART in autobauding mode (0x2) or to the UART in standard mode (0x0).

Use limitation:

- Only 7- and 8-bit characters (5- and 6-bit not supported)
- 7-bit character with space parity not supported
- Baud rate between 1200 and 115,200 bps (10 possibilities)

18.3.8.1.3.5 Error Detection

When the UARTi.UART_LSR register is read, the UARTi.UART_LSR[4:2] bit field reflects the error bits (BI: break condition, FE: framing error, PE: parity error) of the character at the top of the RX FIFO (the next character to be read). Therefore, reading the UARTi.UART_LSR register and then reading the UARTi.UART_RHR register identifies errors in a character.

Reading the UARTi.UART_RHR register updates the BI, FE, and PE bits (see Table 18-11 for the UART mode interrupts).

The UARTi.UART_LSR[7] RX_FIFO_STS bit is set when there is an error in the RX FIFO and is cleared only when no errors remain in the RX FIFO.

NOTE: Reading the UARTi.UART_LSR register does not cause an increment of the RX FIFO read pointer. The RX FIFO read pointer is incremented by reading the UARTi.UART_RHR register.

Reading the UARTi.UART_LSR register clears the OE bit if it is set (see Table 18-11 for the UART mode interrupts).

18.3.8.1.3.6 Overrun During Receive

Overrun during receive occurs if the RX state-machine tries to write data into the RX FIFO when it is already full. When overrun occurs, the device interrupts the MPU with the UARTi.UART_IIR[5:1] IT_TYPE bit field set to 0x3 (receiver line status error) and discards the remaining portion of the frame.

Overrun also causes an internal flag to be set, which disables further reception. Before the next frame can be received, the MPU must:

- Reset the RX FIFO.
- Read the UARTi.UART_RESUME register, which clears the internal flag.

18.3.8.1.3.7 Time-Out and Break Conditions

18.3.8.1.3.7.1 Time-Out Counter

An RX idle condition is detected when the receiver line (uarti_rx) is high for a time that equals 4x the programmed word length + 12 bits. uarti_rx is sampled midway through each bit.

For sleep mode, the counter is reset when there is activity on uarti_rx.

For the time-out interrupt, the counter counts only when there is data in the RX FIFO, and the count is reset when there is activity on uarti_rx or when the UARTi.UART_RHR register is read.

Copyright © 2011, Texas Instruments Incorporated

18.3.8.1.3.7.2 Break Condition

When a break condition occurs, uarti_tx is pulled low. A break condition is activated by setting the UARTi.UART_LCR[6] BREAK_EN bit. The break condition is not aligned on word stream (a break condition can occur in the middle of a character). The only way to send a break condition on a full character is:

1. Reset the TX FIFO (if enabled).
2. Wait for the transmit shift register to empty (the UARTi.UART_LSR[6] TX_SR_E bit is set to 1).
3. Take a guard time according to stop-bit definition.
4. Set the BREAK_EN bit to 1.

The break condition is asserted while the BREAK_EN bit is set to 1.

The time-out counter and break condition apply only to UART modem operation and not to IrDA/CIR mode operation.

18.3.8.2 IrDA Mode (UART3 Only)

18.3.8.2.1 IrDA Clock Generation: Baud Generator

The IrDA function contains a programmable baud generator and a set of fixed dividers that divide the 48-MHz clock input down to the expected baud rate.

Figure 18-15 shows the baud rate generator and associated controls.

Figure 18-15. Baud Rate Generator

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

CAUTION

Before initializing or modifying clock parameter controls (UARTi.UART_DLH, UARTi.UART_DLL), MODE_SELECT=DISABLE (UARTi.UART_MDR1[2:0]) must be set to 0x7). Failure to observe this rule can result in unpredictable module behavior.

18.3.8.2.2 Choosing the Appropriate Divisor Value

Three divisor values are:

- SIR mode: Divisor value = Operating frequency/(16x baud rate)
- MIR mode: Divisor value = Operating frequency/(41x/42x baud rate)
- FIR mode: Divisor value = None

Table 18-28 lists the IrDA baud rate settings.

Table 18-28. IrDA Baud Rate Settings

Baud Rate	IR Mode	Baud Multiple	Encoding	DLH, DLL (Decimal)	Actual Baud Rate	Error (%)	Source Jitter (%)	Pulse Duration
2.4 kbps	SIR	16x	3/16	1250	2.4 kbps	0	0	78.1 µs
9.6 kbps	SIR	16x	3/16	312	9.6153 kbps	+0.16	0	19.5 µs
19.2 kbps	SIR	16x	3/16	156	19.231 kbps	+0.16	0	9.75 µs
38.4 kbps	SIR	16x	3/16	78	38.462 kbps	+0.16	0	4.87 µs
57.6 kbps	SIR	16x	3/16	52	57.692 kbps	+0.16	0	3.25 µs
115.2 kbps	SIR	16x	3/16	26	115.38 kbps	+0.16	0	1.62 µs
0.576 Mbps	MIR	41x/42x	1/4	2	0.5756 Mbps[(1)]	0	+1.63/–0.80	416 ns
1.152 Mbps	MIR	41x/42x	1/4	1	1.1511 Mbps[(1)]	0	+1.63/–0.80	208 ns
4 Mbps	FIR	6x	4 PPM	–	4 Mbps	0	0	125 ns

(1) Average value

NOTE: Baud rate error and source jitter table values do not include 48-MHz reference clock error and jitter.

18.3.8.2.3 IrDA Data Formatting

The methods described in this section apply to all IrDA modes (SIR, MIR, and FIR).

18.3.8.2.3.1 IR RX Polarity Control

The UART3.UART_MDR2[6] IRRXINVERT bit provides the flexibility to invert the uart3_rx_irrx pin in the UART to ensure that the protocol at the output of the transceiver has the same polarity at module level. By default, the uart3_rx_irrx pin is inverted because most transceivers invert the IR receive pin.

18.3.8.2.3.2 IrDA Reception Control

The module can transmit and receive data, but when the device is transmitting, the IR RX circuitry is automatically disabled by hardware.

Operation of the uart3_rx_irrx input can be disabled by the UART3.UART_ACREG[5] DIS_IR_RX bit.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.3.8.2.3.3 IR Address Checking

In all IR modes, when address checking is enabled, only frames intended for the device are written to the RX FIFO. This restriction avoids receiving frames not meant for this device in a multipoint infrared environment. It is possible to program two frame addresses that the UART IrDA receives, with the UART3.UART_XON1_ADDR1[7:0] XON_WORD1 and UART3.UART_XON2_ADDR2[7:0] XON_WORD2 bit fields.

Setting the UART_EFR[0] bit to 1 selects address1 checking. Setting the UART_EFR[1] bit to 1 selects address2 checking. Setting the UART_EFR[1:0] bit field to 0 disables all address checking operations. If both bits are set, the incoming frame is checked for private and public addresses.

If address checking is disabled, all received frames write to the RX FIFO.

18.3.8.2.3.4 Frame Closing

A transmission frame can be terminated in two ways:

- Frame-length method: Set the UART3.UART_MDR1[7] FRAME_END_MODE bit to 0. The MPU writes the value of the frame length to the UART3.UART_TXFLH and UART3.UART_TXFLL registers. The device automatically attaches end flags to the frame when the number of bytes transmitted equals the value of the frame length.
- Set-EOT bit method: Set the FRAME_END_MODE bit to 1. The MPU writes 1 to the UART3.UART_ACREG[0] EOT bit just before it writes the last byte to the TX FIFO. When the MPU writes the last byte to the TX FIFO, the device internally sets the tag bit for that character in the TX FIFO. As the TX state-machine reads data from the TX FIFO, it uses this tag-bit information to attach end flags and correctly terminate the frame.

18.3.8.2.3.5 Store and Controlled Transmission

In store and controlled transmission (SCT) mode, the MPU starts writing data to the TX FIFO. Then, after writing a part of a frame (for a bigger frame) or an entire frame (a small frame; that is, a supervisory frame), the MPU writes 1 to the UART3.UART_ACREG[2] SCTX_EN bit (deferred TX start) to start transmission.

SCT mode is enabled by setting the UART3.UART_MDR1[5] SCT bit to 1. This transmission method differs from normal mode, in which data transmission starts immediately after data is written to the TX FIFO. SCT mode is useful for sending short frames without TX underrun.

18.3.8.2.3.6 Error Detection

When the UART3.UART_LSR register is read, the UART3.UART_LSR[4:2] bit field reflects the error bits [FL, CRC, ABORT] of the frame at the top of the STATUS FIFO (the next frame status to be read).

The error is triggered by an interrupt (for IrDA mode interrupts, see Table 18-12). The STATUS FIFO must be read until empty (a maximum of eight reads is required).

18.3.8.2.3.7 Underrun During Transmission

Underrun during transmission occurs when the TX FIFO is empty before the end of the frame is transmitted. When underrun occurs, the device closes the frame with end flags but attaches an incorrect CRC value. The receiving device detects a CRC error and discards the frame; it can then ask for a retransmission.

Underrun also causes an internal flag to be set, which disables additional transmissions. Before the next frame can be transmitted, the MPU must:

- Reset the TX FIFO.
- Read the UART3.UART_RESUME register, which clears the internal flag.

This function can be disabled by the UART3.UART_ACREG[4] DIS_TX_UNDERRUN bit, compensated by the extension of the stop-bit in transmission if the TX FIFO is empty.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.3.8.2.3.8 Overrun During Receive

Overrun during receive for the IrDA mode has the same function as that for the UART mode (see Section 18.3.8.1.3.6, *Overrun During Receive*).

18.3.8.2.3.9 Status FIFO

In IrDA modes, a status FIFO records the received frame status. When a complete frame is received, the length of the frame and the error bits associated with the frame are written to the status FIFO.

Reading the UART3.UART_SFREGH[3:0] MSB and UART3.UART_SFREGL[3:0] (LSB) bit fields obtains the frame length. The frame error status is read in the UART3.UART_SFLSR register. Reading the UART3.UART_SFLSR register increments the status FIFO read pointer. Because the status FIFO is eight entries deep, it can hold the status of eight frames.

The MPU uses the frame-length information to locate the frame boundary in the received frame data. The MPU can screen bad frames using the error status information and can later request the sender to resend only the bad frames.

This status FIFO can be used effectively in DMA mode because the MPU must be interrupted only when the programmed status FIFO trigger level is reached, not each time a frame is received.

18.3.8.2.4 SIR Mode Data Formatting

This section provides specific instructions for SIR mode programming.

18.3.8.2.4.1 Abort Sequence

The transmitter can prematurely close a frame (abort) by sending the sequence 0x7DC1. The abort pattern closes the frame without a CRC field or an ending flag.

A transmission frame can be aborted by setting the UART3.UART_ACREG[1] ABORT_EN bit to 1. When this bit is set to 1, 0x7D and 0xC1 are transmitted and the frame is not terminated with CRC or stop flags.

When a 0x7D character followed immediately by a 0xC1 character is received without transparency, the receiver treats a frame as an aborted frame.

CAUTION

When the TX FIFO is not empty and the UART3.UART_MDR1[5] SCT bit is set to 1, the UART IrDA starts a new transfer with data of a previous frame when the aborted frame is sent. Therefore, the TX FIFO must be reset before sending an aborted frame.

18.3.8.2.4.2 Pulse Shaping

SIR mode supports the 3/16 or the 1.6-µs pulse duration methods in receive and transmit. The UART3.UART_ACREG[7] PULSE_TYPE bit selects the pulse width method in the transmit mode.

18.3.8.2.4.3 SIR Free Format Programming

The SIR FF mode is selected by setting the module in the UART mode (UART3.UART_MDR1[2:0] MODE_SELECT = 0x0) and the UART3.UART_MDR2[3] UART_PULSE bit to 1 to allow pulse shaping.

Because the bit format stays the same, some UART mode configuration registers must be set at specific values:

- UART3.UART_LCR[1:0] CHAR_LENGTH bit field = 0x3 (8 data bits)
- UART3.UART_LCR[2] NB_STOP bit = 0x0 (1 stop-bit)
- UART3.UART_LCR[3] PARITY_EN bit = 0x0 (no parity)

The UART mode interrupts are used for the SIR FF mode, but many are not relevant (XOFF, RTS, CTS, modem status register, etc.).

Copyright © 2011, Texas Instruments Incorporated

18.3.8.2.5 MIR and FIR Mode Data Formatting

This section describes common instructions for FIR and MIR mode programming.

At the end of a frame reception, the MPU reads the line status register (UART3.UART_LSR) to detect errors in the received frame.

When the UART3.UART_MDR1[6] SIP_MODE bit is set to 1, the TX state-machine always sends one SIP at the end of a transmission frame. However, when the SIP_MODE bit is set to 0, SIP transmission depends on the UART3.UART_ACREG[3] SEND_SIP bit.

The MPU can set the SEND_SIP bit at least once every 500 ms. The advantage of this approach over the default approach is that the TX state-machine does not have to send the SIP at the end of each frame, thus reducing the overhead required.

18.3.8.3 CIR Mode (UART3 Only)

18.3.8.3.1 CIR Mode Clock Generation

Depending on the encoding method (variable pulse distance/biphase), the MPU must develop a data structure that combines 1 and 0 with a *t* period to encode the complete frame to transmit. This can then be transmitted to the infrared output with a modulation method, as shown in Figure 18-16.

Figure 18-16. CIR Mode Block Components

Based on the requested modulation frequency, the UART3.UART_CFPS register must be set with the correct dividing value to provide an accurate pulse frequency:

Dividing value = (FCLK/12)/MODfreq

Where:

FCLK = System clock frequency (48 MHz)

12 = Real value of baud multiple

MODfreq = Effective frequency of the modulation (MHz)

Example: For a targeted modulation frequency of 36 kHz, the value of UART_CFPS must be set to 0x7 (decimal), which provides a modulation frequency of 36.04 kHz.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

NOTE: The UART3.UART_CFPS register starts with a reset value of 105 (decimal), which translates to a frequency of 38.1 kHz.

The duty cycle of these pulses is user-defined by the pulse duty register bits in the UART3.UART_MDR2 register. Table 18-29 shows the duty cycle.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 18-29. Duty Cycle

UART_MDR2[5:4]	Duty Cycle (High-Level)
00	1/4
01	1/3
10	5/12
11	1/2

18.3.8.3.2 CIR Data Formatting

The methods described in this section apply to all CIR modes.

18.3.8.3.2.1 IR RX Polarity Control

The IR RX polarity control for CIR mode has the same function as that for IrDA mode (see Section 18.3.8.2.3.1, *IR RX Polarity Control*).

18.3.8.3.2.2 CIR Transmission

In transmission, the MPU software must exercise an element of real-time control to transmit data packets, each of which must be emitted at a constant delay from the start-bits of each individual packet. Thus, when sending a series of packets, the packet-to-packet delay must respect a specific delay. Two methods can be used to control this delay:

- Filling the TX FIFO with a number of zero bits that are transmitted with a *t* period
- Using an external system timer to control the delay between each start-of-frame or between the end of a frame and the start of the next one. This can be performed by:
  - Controlling the start of the frame using the UART3.UART_MDR1[5] SCT bit and the UART3.UART_ACREG[2] SCTX_EN bit, depending on the timer status
  - Using the UART3.UART_IIR[5] TX_STATUS_IT interrupt bit to preload the next frame in the TX FIFO and to control the start of the timer (in case of control delay between the end of a frame and the start of the next frame)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.4 UART/IrDA/CIR Basic Programming Model

18.4.1 UART Programming Model

18.4.1.1 Quick Start

This section describes the procedure for operating the UART with FIFO and DMA or interrupts. This three-part procedure ensures the quick start of the UART. It does not cover every UART feature.

The first programming model covers software reset of the UART. The second programming model describes FIFO and DMA configuration. The last programming model describes protocol, baud rate, and interrupt configuration.

NOTE: Each programming model can be used independently of the other two; for instance, reconfiguring the FIFOs and DMA settings only.

Each programming model can be executed starting from any UART register access mode (register modes, submodes, and other register dependencies). However, if the UART register access mode is known before executing the programming model, some steps that enable or restore register access are optional. For more information, see Section 18.3.7.1, *Register Access Modes*.

18.4.1.1.1 Software Reset

To clear the UART registers, perform the following steps:

1. Initiate a software reset:
   Set the UARTi.UART_SYSC[1] SOFTRESET bit to 1.
2. Wait for the end of the reset operation:
   Poll the UARTi.UART_SYSS[0] RESETDONE bit until it equals 1.

18.4.1.1.2 FIFOs and DMA Settings

To enable and configure the receive and transmit FIFOs and program the DMA mode, perform the following steps:

1. Switch to register configuration mode B to access the UARTi.UART_EFR register:
   (a) Save the current UARTi.UART_LCR register value.
   (b) Set the UARTi.UART_LCR register value to 0x00BF.
2. Enable register submode TCR_TLR to access the UARTi.UART_TLR register (part 1 of 2):
   (a) Save the UARTi.UART_EFR[4] ENHANCED_EN value.
   (b) Set the UARTi.UART_EFR[4] ENHANCED_EN bit to 1.
3. Switch to register configuration mode A to access the UARTi.UART_MCR register:
   Set the UARTi.UART_LCR register value to 0x0080.
4. Enable register submode TCR_TLR to access the UARTi.UART_TLR register (part 2 of 2):
   (a) Save the UARTi.UART_MCR[6] TCR_TLR value.
   (b) Set the UARTi.UART_MCR[6] TCR_TLR bit to 1.
5. Enable the FIFO; load the new FIFO triggers (part 1 of 3) and the new DMA mode (part 1 of 2):
   Set the following bits to the desired values:
   - UARTi.UART_FCR[7:6] RX_FIFO_TRIG
   - UARTi.UART_FCR[5:4] TX_FIFO_TRIG
   - UARTi.UART_FCR[3] DMA_MODE
   - UARTi.UART_FCR[0] FIFO_ENABLE (0: Disable the FIFO; 1: Enable the FIFO)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

NOTE: The UARTi.UART_FCR register is not readable.

6. Switch to register configuration mode B to access the UARTi.UART_EFR register:
   Set the UARTi.UART_LCR register value to 0x00BF.
7. Load the new FIFO triggers (part 2 of 3):
   Set the following bits to the desired values:
   - UARTi.UART_TLR[7:4] RX_FIFO_TRIG_DMA
   - UARTi.UART_TLR[3:0] TX_FIFO_TRIG_DMA
8. Load the new FIFO triggers (part 3 of 3) and the new DMA mode (part 2 of 2):
   Set the following bits to the desired values:
   - UARTi.UART_SCR[7] RX_TRIG_GRANU1
   - UARTi.UART_SCR[6] TX_TRIG_GRANU1
   - UARTi.UART_SCR[2:1] DMA_MODE_2
   - UARTi.UART_SCR[0] DMA_MODE_CTL
9. Restore the UARTi.UART_EFR[4] ENHANCED_EN value saved in Step 2a.
10. Switch to register configuration mode A to access the UARTi.UART_MCR register:
    Set the UARTi.UART_LCR register value to 0x0080.
11. Restore the UARTi.UART_MCR[6] TCR_TLR value saved in Step 4a.
12. Restore the UARTi.UART_LCR value saved in Step 1a.

Triggers are used to generate interrupt and DMA requests. See Section 18.3.6.1.1, *Transmit FIFO Trigger,* to choose the following values:

- UARTi.UART_FCR[5:4] TX_FIFO_TRIG
- UARTi.UART_TLR[3:0] TX_FIFO_TRIG_DMA
- UARTi.UART_SCR[6] TX_TRIG_GRANU1

Triggers are used to generate interrupt and DMA requests. See Section 18.3.6.1.2, *Receive FIFO Trigger*, to choose the following values:

- UARTi.UART_FCR[7:6] RX_FIFO_TRIG
- UARTi.UART_TLR[7:4] RX_FIFO_TRIG_DMA
- UARTi.UART_SCR[7] RX_TRIG_GRANU1

DMA mode enables DMA requests. See Section 18.3.6.4, *FIFO DMA Mode Operation*, to choose the following values:

- UARTi.UART_FCR[3] DMA_MODE
- UARTi.UART_SCR[2:1] DMA_MODE_2
- UARTi.UART_SCR[0] DMA_MODE_CTL

18.4.1.1.3 Protocol, Baud Rate, and Interrupt Settings

To program the protocol, baud rate, and interrupt settings, perform the following steps:

1. Disable UART to access the UARTi.UART_DLL and UARTi.UART_DLH registers:
   Set the UARTi.UART_MDR1[2:0] MODE_SELECT bit field to 0x7.
2. Switch to register configuration mode B to access the UARTi.UART_EFR register:
   Set the UARTi.UART_LCR register value to 0x00BF.
3. Enable access to the UARTi.UART_IER[7:4] bit field:
   (a) Save the UARTi.UART_EFR[4] ENHANCED_EN value.
   (b) Set the UARTi.UART_EFR[4] ENHANCED_EN bit to 1.
4. Switch to register operational mode to access the UARTi.UART_IER register:
   Set the UARTi.UART_LCR register value to 0x0000.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

5. Clear the UARTi.UART_IER register (set the UARTi.UART_IER[4] SLEEP_MODE bit to 0 to change the UARTi.UART_DLL and UARTi.UART_DLH registers). Set the UARTi.UART_IER register value to 0x0000.
6. Switch to register configuration mode B to access the UARTi.UART_DLL and UARTi.UART_DLH registers:

   Set the UARTi.UART_LCR register value to 0x00BF.
7. Load the new divisor value:

   Set the UARTi.UART_DLL[7:0] CLOCK_LSB and UARTi.UART_DLH[5:0] CLOCK_MSB bit fields to the desired values.
8. Switch to register operational mode to access the UARTi.UART_IER register:

   Set the UARTi.UART_LCR register value to 0x0000.
9. Load the new interrupt configuration (0: Disable the interrupt; 1: Enable the interrupt):

   Set the following bits to the desired values:
   - UARTi.UART_IER[7] CTS_IT
   - UARTi.UART_IER[6] RTS_IT
   - UARTi.UART_IER[5] XOFF_IT
   - UARTi.UART_IER[4] SLEEP_MODE
   - UARTi.UART_IER[3] MODEM_STS_IT
   - UARTi.UART_IER[2] LINE_STS_IT
   - UARTi.UART_IER[1] THR_IT
   - UARTi.UART_IER[0] RHR_IT
10. Switch to register configuration mode B to access the UARTi.UART_EFR register:

    Set the UARTi.UART_LCR register value to 0x00BF.
11. Restore the UARTi.UART_EFR[4] ENHANCED_EN value saved in Step 3a.
12. Load the new protocol formatting (parity, stop-bit, character length) and switch to register operational mode:

    Set the UARTi.UART_LCR[7] DIV_EN bit to 0.

    Set the UARTi.UART_LCR[6] BREAK_EN bit to 0.

    Set the following bits to the desired values:
    - UARTi.UART_LCR[5] PARITY_TYPE_2
    - UARTi.UART_LCR[4] PARITY_TYPE_1
    - UARTi.UART_LCR[3] PARITY_EN
    - UARTi.UART_LCR[2] NB_STOP
    - UARTi.UART_LCR[1:0] CHAR_LENGTH
13. Load the new UART mode:

    Set the UARTi.UART_MDR1[2:0] MODE_SELECT bit field to the desired value.

See Section 18.3.8.1.2, *Choosing the Appropriate Divisor Value*, to choose the following values:

- UARTi.UART_DLL[7:0] CLOCK_LSB
- UARTi.UART_DLH[5:0] CLOCK_MSB
- UARTi.UART_MDR1[2:0] MODE_SELECT

See Section 18.3.8.1.3.1, *Frame Formatting*, to choose the following values:

- UARTi.UART_LCR[5] PARITY_TYPE_2
- UARTi.UART_LCR[4] PARITY_TYPE_1
- UARTi.UART_LCR[3] PARITY_EN
- UARTi.UART_LCR[2] NB_STOP
- UARTi.UART_LCR[1:0] CHAR_LENGTH

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.4.1.2 Hardware and Software Flow Control Configuration

This section describes the programming steps to enable and configure hardware and software flow control. Hardware and software flow control cannot be used at the same time.

NOTE: Each programming model can be executed starting from any UART register access mode (register modes, submodes, and other register dependencies). However, if the UART register access mode is known before executing the programming model, some steps that enable or restore register access are optional. For more information, see Section 18.3.7.1, *Register Access Modes*.

18.4.1.2.1 Hardware Flow Control Configuration

To enable and configure hardware flow control, perform the following steps:

1. Switch to register configuration mode A to access the UARTi.UART_MCR register:
   (a) Save the current UARTi.UART_LCR register value.
   (b) Set the UARTi.UART_LCR register value to 0x0080.
2. Enable register submode TCR_TLR to access the UARTi.UART_TCR register (part 1 of 2):
   (a) Save the UARTi.UART_MCR[6] TCR_TLR value.
   (b) Set the UARTi.UART_MCR[6] TCR_TLR bit to 1.
3. Switch to register configuration mode B to access the UARTi.UART_EFR register:
   Set the UARTi.UART_LCR register value to 0x00BF.
4. Enable register submode TCR_TLR to access the UARTi.UART_TCR register (part 2 of 2):
   (a) Save the UARTi.UART_EFR[4] ENHANCED_EN value.
   (b) Set the UARTi.UART_EFR[4] ENHANCED_EN bit to 1.
5. Load the new start and halt trigger values for hardware flow control:
   Set the following bits to the desired values:
   - UARTi.UART_TCR[7:4] AUTO_RTS_START
   - UARTi.UART_TCR[3:0] AUTO_RTS_HALT
6. Enable or disable receive and transmit hardware flow control mode and restore the UARTi.UART_EFR[4] ENHANCED_EN value saved in Step 4a.
   Set the following bits to the desired values:
   - UARTi.UART_EFR[7] AUTO_CTS_EN (0: Disable; 1: Enable)
   - UARTi.UART_EFR[6] AUTO_RTS_EN (0: Disable; 1: Enable)

   Restore the UARTi.UART_EFR[4] ENHANCED_EN bit to the saved value.
7. Switch to register configuration mode A to access the UARTi.UART_MCR register:
   Set the UARTi.UART_LCR register value to 0x0080.
8. Restore the UARTi.UART_MCR[6] TCR_TLR value saved in Step 2a.
9. Restore the UARTi.UART_LCR value saved in Step 1a.

See Section 18.3.8.1.3.2, *Hardware Flow Control*, to choose the following values:

- UARTi.UART_EFR[7] AUTO_CTS_EN
- UARTi.UART_EFR[6] AUTO_RTS_EN
- UARTi.UART_TCR[7:4] AUTO_RTS_START
- UARTi.UART_TCR[3:0] AUTO_RTS_HALT

18.4.1.2.2 Software Flow Control Configuration

To enable and configure software flow control, perform the following steps:

1. Switch to register configuration mode B to access the UARTi.UART_EFR register.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

(a) Save the current UARTi.UART_LCR register value.

(b) Set the UARTi.UART_LCR register value to 0x00BF.

2. Enable register submode XOFF to access the UARTi.UART_XOFF1 and UARTi.UART_XOFF2 registers:

   (a) Save the UARTi.UART_EFR[4] ENHANCED_EN value.

   (b) Set the UARTi.UART_EFR[4] ENHANCED_EN bit to 0.

3. Load the new software flow control characters:

   Set the following bits to the desired values:

   - UARTi.UART_XON1_ADDR1[7:0] XON_WORD1
   - UARTi.UART_XON2_ADDR2[7:0] XON_WORD2
   - UARTi.UART_XOFF1[7:0] XOFF_WORD1
   - UARTi.UART_XOFF2[7:0] XOFF_WORD2

4. Enable access to the UARTi.UART_MCR[7:5] bit field and enable register submode TCR_TLR to access the UARTi.UART_TCR register (part 1 of 2):

   Set the UARTi.UART_EFR[4] ENHANCED_EN bit to 1.

5. Switch to register configuration mode A to access the UARTi.UART_MCR register:

   Set the UARTi.UART_LCR register value to 0x0080.

6. Enable register submode TCR_TLR to access the UARTi.UART_TCR register (part 2 of 2) and enable or disable XON any function:

   (a) Save the UARTi.UART_MCR[6] TCR_TLR value.

   (b) Set the UARTi.UART_MCR[6] TCR_TLR bit to 1.

   (c) Set the UARTi.UART_MCR[5] XON_EN bit to the desired value (0: Disable; 1: Enable).

7. Switch to register configuration mode B to access the UARTi.UART_EFR register:

   Set the UARTi.UART_LCR register value to 0x00BF.

8. Load the new start and halt trigger values for software flow control:

   Set the following bits to the desired values:

   - UARTi.UART_TCR[7:4] AUTO_RTS_START
   - UARTi.UART_TCR[3:0] AUTO_RTS_HALT

9. Enable or disable special character function and load the new software flow control mode and restore the UARTi.UART_EFR[4] ENHANCED_EN value saved in Step 2a:

   Set the following bits to the desired values:

   - UARTi.UART_EFR[5] SPEC_CHAR (0: Disable; 1: Enable)
   - UARTi.UART_EFR[3:0] SW_FLOW_CONTROL

   Restore the UARTi.UART_EFR[4] ENHANCED_EN bit to the saved value.

10. Switch to register configuration mode A to access the UARTi.UART_MCR register:

    Set the UARTi.UART_LCR register value to 0x0080.

11. Restore the UARTi.UART_MCR[6] TCR_TLR bit value saved in Step 6a.

12. Restore the UARTi.UART_LCR value saved in Step 1a.

See Section 18.3.8.1.3.3, *Software Flow Control*, to choose the following values:

- UARTi.UART_EFR[5] SPEC_CHAR
- UARTi.UART_EFR[3:0] SW_FLOW_CONTROL
- UARTi.UART_TCR[7:4] AUTO_RTS_START
- UARTi.UART_TCR[3:0] AUTO_RTS_HALT
- UARTi.UART_XON1_ADDR1[7:0] XON_WORD1
- UARTi.UART_XON2_ADDR2[7:0] XON_WORD2
- UARTi.UART_XOFF1[7:0] XOFF_WORD1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

- UARTi.UART_XOFF2[7:0] XOFF_WORD2

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.4.2 IrDA Programming Model (UART3 Only)

18.4.2.1 SIR Mode

18.4.2.1.1 Receive

The following programming model explains how to program the module to receive an IrDA frame with parity forced to 1, baud rate = 112.5KB, FIFOs disabled, 2 stop-bits, and 8-bit word length:

1. Disable the UART before accessing the UARTi.UART_DLL and UARTi.UART_DLH registers:
   Set the UARTi.UART_MDR1[2:0] MODE_SELECT bit field to 0x7.
2. Grant access to the UART_DLL and UART_DLH registers (the UART_LCR[7] DIV_EN bit = 1):
   UART3.UART_LCR = 0x80 (Data format is unaffected by the use and settings of the UART3.UART_LCR register in IrDA mode.)
3. Load the new baud rate (115.2 kbps):
   UART3.UART_DLL = 0x1A
   UART3.UART_DLH = 0x00
4. Set SIR mode:
   UART3.UART_MDR1[2:0] MODE_SELECT = 0x1
5. Disable access to the UART_DLL and UART_DLH registers and switch to register operational mode:
   UART3.UART_LCR = 0x00.
6. Optional: Enable the RHR interrupt:
   UART3.UART_IER[0] RHR_IT = 0x1

18.4.2.1.2 Transmit

The following programming model explains how to program the module to transmit an IrDA 6-byte frame with no parity, baud rate = 112.5 kbps, FIFOs disabled, 3/16 encoding, 2 stop-bits, and 7-bit word length:

1. Disable the UART before accessing the UARTi.UART_DLL and UARTi.UART_DLH registers:
   Set the UART_MDR1[2:0] MODE_SELECT bit field to 0x7.
2. Grant access to the UART_EFR register:
   UART3.UART_LCR = 0xBF
3. Enable the enhanced features (the UART_EFR[4] ENAHNCED_EN bit = 1):
   Set the UART3.UART_EFR register value to 0x10.
4. Grant access to the UART_DLL and UART_DLH registers (the UART_LCR[7] DIV_EN bit = 1):
   UART3.UART_LCR = 0x80 (Data format is unaffected by the use and settings of the UART3.UART_LCR register in IrDA mode.)
5. Load the new baud rate (115.2 kbps):
   UART3.UART_DLL = 0x1A
   UART3.UART_DLH = 0x00
6. Set SIR mode (the UART_MDR1[2:0] MODE_SELECT bit field = 0x1):
   UART3.UART_MDR1 = 0x01
7. Disable access to the UART_DLL and UART_DLH registers and switch to register operational mode:
   UART3.UART_LCR = 0x00.
8. Force DTR output to active:
   UART3.UART_MCR[0] DTR = 1
9. Optional: Enable the THR interrupt:
   UART3.UART_IER[1] THR_IT = 1
10. Set transmit frame length to 6 bytes:
    UART3.UART_TXFLL = 0x06
11. Set 7 starts of frame transmission:
    UART3.UART_EBLR = 0x08
12. Optional: Set SIR pulse width to be 1.6 us:
    UART3.UART_ACREG[7] PULSE_TYPE = 1
13. Load the UART_THR register with the data to be transmitted.

Copyright © 2011, Texas Instruments Incorporated

18.4.2.2 MIR Mode

18.4.2.2.1 Receive

The following programming model explains how to program the module to receive an IrDA frame with no parity, baud rate = 1.152 Mpbs, and FIFOs disabled.

1. Disable the UART before accessing the UARTi.UART_DLL and UARTi.UART_DLH registers:
   Set the UARTi.UART_MDR1[2:0] MODE_SELECT bit field to 0x7.
2. Grant access to the UART_DLL and UART_DLH registers (UART_LCR[7] DIV_EN bit = 1):
   UART3.UART_LCR = 0x80 (Data format is unaffected by the use and settings of the UART3.UART_LCR register in IrDA mode.)
3. Load the new baud rate (1.152 Mpbs):
   UART3.UART_DLL = 0x01
   UART3.UART_DLH = 0x00
4. Set MIR mode:
   UART3.UART_MDR1[2:0] MODE_SELECT = 0x4
5. Disable access to the UART_DLL and UART_DLH registers and switch to register operational mode:
   UART3.UART_LCR = 0x00
6. Force DTR output to active (UART_MCR[0] DTR = 1):
   Force RTS output to active (UART_MCR[1] RTS = 1).
   UART3.UART_MCR = 0x3
7. Optional: Enable the RHR interrupt:
   UART3.UART_IER[0] RHR_IT = 1

18.4.2.2.2 Transmit

The following programming model explains how to program the module to transmit an IrDA 60-byte frame with no parity, baud rate = 1.152 Mpbs, and FIFOs disabled:

1. Disable the UART before accessing the UARTi.UART_DLL and UARTi.UART_DLH registers:
   Set the UARTi.UART_MDR1[2:0] MODE_SELECT bit field to 0x7.
2. Grant access to the UART_DLL and UART_DLH registers (UART_LCR[7] DIV_EN bit = 1):
   UART3.UART_LCR = 0x80 (Data format is unaffected by the use and settings of the UART3.UART_LCR register in IrDA mode.)
3. Load the new baud rate (1.152 Mpbs):
   UART3.UART_DLL = 0x01
   UART3.UART_DLH = 0x00
4. Set MIR mode:
   UART3.UART_MDR1[2:0] MODE_SELECT = 0x4
5. Disable access to the UART_DLL and UART_DLH registers and switch to register operational mode:
   UART3.UART_LCR = 0x00
6. Force DTR output to active:
   UART3.UART_MCR[0] DTR = 1
7. Optional: Enable the THR interrupt:
   UART3.UART_IER[1] THR_IT = 1
8. Set the frame length to 60 bytes:
   UART3.UART_TXFLL = 0x3C
9. Optional: Transmit eight additional starts of frame (MIR mode requires two starts):
   UART3.UART_EBLR = 0x08
10. SIP is sent at the end of transmission:
    UART3.UART_ACREG[3] = 1
11. Load the UART_THR register with the data to be transmitted.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.4.2.3 FIR Mode

18.4.2.3.1 Receive

The following programming model explains how to program the module to receive the IrDA frame with no parity, baud rate = 4 Mbps, FIFOs enabled, 8-bit word length.

1. Disable the UART before accessing the UART3.UART_DLL and UART3.UART_DLH registers:
   Set the UART3.UART_MDR1[2:0] MODE_SELECT bit field to 0x7.
2. Grant access to the UART_DLL and UART_DLH registers (UART_LCR[7] DIV_EN bit = 1):
   UART3.UART_LCR = 0x80 (Data format is unaffected by the use and settings of the UART3.UART_LCR register in IrDA mode.)
3. FIFO clear and enable:
   UART3.UART_FCR = 0x7 (TX/RX FIFO trigger: UART_FCR[7:6] and UART_FCR[5:4])
   UART3.UART_LCR[7] = 0
4. Set FIR mode:
   UART3.UART_MDR1[2:0] MODE_SELECT = 0x5
5. Set frame length:
   UART3.UART_RXFLL = 0xA (Data + CRC + STOP)
6. Disable access to the UART3.UART_DLL registers and UART3.UART_DLH and switch to register operational mode:
   UART3.UART_LCR[7] DIV_EN = 0x0
7. Optional: Enable the RHR interrupt:
   UART3.UART_IER[0] RHR_IT = 1

18.4.2.3.2 Transmit

The following programming model explains how to program the module to transmit an IrDA 4-byte frame with no parity, baud rate = 4 Mbps, FIFOs enabled, and 8-bit word length.

1. Disable the UART before accessing the UART3.UART_DLL and UART3.UART_DLH registers:
   Set the UART3.UART_MDR1[2:0] MODE_SELECT bit field to 0x7.
2. Grant access to EFR_REG:
   UART3.UART_LCR = 0xBF
3. Enable the enhanced features (EFR_REG[4] ENAHNCED_EN = 0x1):
   UART3.UART_EFR = 0x10
4. FIFO clear and enable:
   UART3.UART_FCR = 0x7 (TX/RX FIFO trigger: UART_FCR[7:6] and UART_FCR[5:4]).
   UART3.UART_LCR[7] = 0
5. Set FIR mode and enable auto-SIP mode:
   UART3.UART_MDR1 = 0x45
6. Set frame length:
   UART3.UART_TXFLL = 0x4
   UART3.UART_TXFLH = 0x0
   UART3.UART_RXFLL = 0xA (Data + CRC + STOP)
   UART3.UART_RXFLH = 0x0
7. Force DTR output to active:
   UART3.UART_MCR[0] DTR = 0x1
8. Optional: Enable the THR interrupt:
   UART3.UART_IER[1] THR_IT = 0x1
9. Optional: Transmit eight additional starts of frame (MIR mode requires two starts):
   UART3.UART_EBLR = 0x08
10. SIP is sent at the end of transmission:
    UART3.UART_ACREG[3] = 1
11. Load the UART_THR register with the data to be transmitted.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5 UART Registers

Each register is selected using a combination of address and for some LCR register bit(s) setting as shown in Table 18-30. The following registers are accessible by the local host (LH) at address = module base address + address offset. The module base address is the module start address. Note that register address offsets depends upon the module address alignment at the system top level.

Table 18-30. UART Registers

Address Offset	Registers					
	LCR[7] = 0		LCR[7:0] ≠ BFh		LCR[7:0] = BFh	
	Read	Write	Read	Write	Read	Write
0h	RHR	THR	DLL	DLL	DLL	DLL
4h	IER[(1)]	IER[(1)]	DLH	DLH	DLH	DLH
8h	IIR	FCR[(2)]	IIR	FCR[(2)]	EFR	EFR
Ch	LCR	LCR	LCR	LCR	LCR	LCR
10h	MCR[(2)]	MCR[(2)]	MCR[(2)]	MCR[(2)]	XON1/ADDR1	XON1/ADDR1
14h	LSR	-	LSR	-	XON2/ADDR2	XON2/ADDR2
18h	MSR/TCR[(3)]	TCR[(3)]	MSR/TCR[(3)]	TCR[(3)]	XOFF1/TCR[(3)]	XOFF1/TCR[(3)]
1Ch	SPR/TLR[(3)]	SPR/TLR[(3)]	SPR/TLR[(3)]	SPR/TLR[(3)]	XOFF2/TLR[(3)]	XOFF2/TLR[(3)]
20h	MDR1	MDR1	MDR1	MDR1	MDR1	MDR1
24h	MDR2	MDR2	MDR2	MDR2	MDR2	MDR2
28h	SFLSR	TXFLL	SFLSR	TXFLL	SFLSR	TXFLL
2Ch	RESUME	TXFLH	RESUME	TXFLH	RESUME	TXFLH
30h	SFREGL	RXFLL	SFREGL	RXFLL	SFREGL	RXFLL
34h	SFREGH	RXFLH	SFREGH	RXFLH	SFREGH	RXFLH
38h	BLR	BLR	UASR	-	UASR	-
3Ch	ACREG	ACREG	-	-	-	-
40h	SCR	SCR	SCR	SCR	SCR	SCR
44h	SSR	SSR[2]	SSR	SSR[2]	SSR	SSR[2]
48h	EBLR	EBLR	-	-	-	-
50h	MVR	-	MVR	-	MVR	-
54h	SYSC	SYSC	SYSC	SYSC	SYSC	SYSC
58h	SYSS	-	SYSS	-	SYSS	-
5Ch	WER	WER	WER	WER	WER	WER
60h	CFPS	CFPS	CFPS	CFPS	CFPS	CFPS

(1) In UART modes, IER[7:4] can only be written when EFR[4] = 1; in IrDA/CIR modes, EFR[4] has no impact on the access to IER[7:4].
(2) MCR[7:5] and FCR[5:4] can only be written when EFR[4] = 1.
(3) Transmission control register (TCR) and trigger level register (TLR) are accessible only when EFR[4] = 1 and MCR[6] = 1.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.1 Receiver Holding Register (RHR)

The receiver section consists of the receiver holding register and the receiver shift register. The RHR is actually a 64-byte FIFO. The receiver shift register receives serial data from RX input. The data is converted to parallel data and moved to the RHR. If the FIFO is disabled, location zero of the FIFO is used to store the single data character.

NOTE: If an overflow occurs, the data in the RHR is not overwritten.

The receiver holding register (RHR) register is shown in Figure 18-17 and described in Table 18-31.

Figure 18-17. Receiver Holding Register (RHR)

15–8	7–0
Reserved	RHR
R-0	R-unknown

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-31. Receiver Holding Register (RHR) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7-0	RHR	0-FFh	Receive holding register.

18.5.2 Transmit Holding Register (THR)

The transmitter section consists of the transmit holding register and the transmit shift register. The transmit holding register is a 64-byte FIFO. The MPU writes data to the THR. The data is placed in the transmit shift register where it is shifted out serially on the TX output. If the FIFO is disabled, location zero of the FIFO is used to store the data.

The transmit holding register (THR) is shown in Figure 18-18 and described in Table 18-32.

Figure 18-18. Transmit Holding Register (THR)

15–8	7–0
Reserved	THR
R-0	W-unknown

LEGEND: R/W = Read/Write; W = Write only; -*n* = value after reset

Table 18-32. Transmit Holding Register (THR) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7-0	THR	0-FFh	Transmit holding register.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.3 Interrupt Enable Register (IER) - UART Mode

The interrupt enable register (IER) can be programmed to enable/disable any interrupt. There are seven types of interrupt in this mode: receiver error, RHR interrupt, THR interrupt, XOFF received and $\overline{CTS}$/$\overline{RTS}$ change of state from low to high. Each interrupt can be enabled/disabled individually. There is also a sleep mode enable bit. The UART interrupt enable register (IER) is shown in Figure 18-19 and described in Table 18-33.

Figure 18-19. UART Interrupt Enable Register (IER)

15							8
Reserved							
R-0							
7	6	5	4	3	2	1	0
CTSIT	RTSIT	XOFFIT	SLEEPMODE	MODEMSTSIT	LINESTSIT	THRIT	RHRIT
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-33. UART Interrupt Enable Register (IER) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7	CTSIT		Can be written only when EFR[4] = 1.
		0	Disables the $\overline{CTS}$ interrupt.
		1	Enables the $\overline{CTS}$ interrupt.
6	RTSIT		Can be written only when EFR[4] = 1.
		0	Disables the $\overline{RTS}$ interrupt.
		1	Enables the $\overline{RTS}$ interrupt.
5	XOFFIT		Can be written only when EFR[4] = 1.
		0	Disables the XOFF interrupt.
		1	Enables the XOFF interrupt.
4	SLEEPMODE		Can be only written when EFR[4] = 1.
		0	Disables sleep mode.
		1	Enables sleep mode (stop baud rate clock when the module is inactive).
3	MODEMSTSIT	0	Disables the modem status register interrupt.
		1	Enables the modem status register interrupt
2	LINESTSIT	0	Disables the receiver line status interrupt.
		1	Enables the receiver line status interrupt.
1	THRIT	0	Disables the THR interrupt.
		1	Enables the THR interrupt.
0	RHRIT	0	Disables the RHR interrupt and time out interrupt.
		1	Enables the RHR interrupt and time out interrupt.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.4 Interrupt Enable Register (IER) - IrDA Mode

The IrDA interrupt enable register (IER) can be programmed to enable/disable any interrupt. There are 8 types of interrupt in these modes, received EOF, LSR interrupt, TX status, status FIFO interrupt, RX overrun, last byte in RX FIFO, THR interrupt, and RHR interrupt. Each interrupt can be enabled/disabled individually. The IrDA interrupt enable register (IER) is shown in Figure 18-20 and described in Table 18-34.

NOTE: The TXSTATUSIT interrupt reflects two possible conditions. The MDR2[0] bit should be read to determine the status in the event of this interrupt.

Figure 18-20. IrDA Interrupt Enable Register (IER)

15							8
Reserved							
R-0							
7	6	5	4	3	2	1	0
EOFIT	LINESTSIT	TXSTATUSIT	STSFIFOTRIGIT	RXOVERRUNIT	LASTRXBYTEIT	THRIT	RHRIT
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-34. IrDA Interrupt Enable Register (IER) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7	EOFIT	0	Disables the received EOF interrupt.
		1	Enables the received EOF interrupt.
6	LINESTSIT	0	Disables the receiver line status interrupt.
		1	Enables the receiver line status interrupt.
5	TXSTATUSIT	0	Disables the TX status interrupt.
		1	Enables the TX status interrupt.
4	STSFIFOTRIGIT	0	Disables status FIFO trigger level interrupt.
		1	Enables status FIFO trigger level interrupt.
3	RXOVERRUNIT	0	Disables the RX overrun interrupt.
		1	Enables the RX overrun interrupt.
2	LASTRXBYTEIT	0	Disables the last byte of frame in RX FIFO interrupt.
		1	Enables the last byte of frame in RX FIFO interrupt.
1	THRIT	0	Disables the THR interrupt.
		1	Enables the THR interrupt.
0	RHRIT	0	Disables the RHR interrupt.
		1	Enables the RHR interrupt.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.5 Interrupt Enable Register (IER) - CIR Mode

The CIR interrupt enable register (IER) can be programmed to enable/disable any interrupt. There are 5 types of interrupt in these modes, TX status, RX overrun, RX stop interrupt, THR interrupt, and RHR interrupt. Each interrupt can be enabled/disabled individually. The CIR interrupt enable register (IER) is shown in Figure 18-21 and described in Table 18-35.

NOTE: In CIR mode, the TXSTATUSIT bit has only one meaning corresponding to the case MDR2[0] = 0.

The RXSTOPIT interrupt is generated based on the value set in the BOF Length register (EBLR).

Figure 18-21. CIR Interrupt Enable Register (IER)

15							8
Reserved							
R-0							
7	6	5	4	3	2	1	0
Reserved		TXSTATUSIT	Reserved	RXOVERRUNIT	RXSTOPIT	THRIT	RHRIT
R-0		R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-35. CIR Interrupt Enable Register (IER) Field Descriptions

Bit	Field	Value	Description
15-6	Reserved	0	Reserved.
5	TXSTATUSIT	0	Disables the TX status interrupt.
		1	Enables the TX status interrupt.
4	Reserved	0	Reserved.
3	RXOVERRUNIT	0	Disables the RX overrun interrupt.
		1	Enables the RX overrun interrupt.
2	RXSTOPIT	0	Disables the RX stop interrupt.
		1	Disables the RX stop interrupt.
1	THRIT	0	Disables the THR interrupt.
		1	Enables the THR interrupt.
0	RHRIT	0	Disables the RHR interrupt.
		1	Enables the RHR interrupt.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.6 Interrupt Identification Register (IIR) - UART Mode

The UART interrupt identification register (IIR) is a read-only register that provides the source of the interrupt. The UART interrupt identification register (IIR) is shown in Figure 18-22 and described in Table 18-36.

NOTE: An interrupt source can be flagged only if enabled in the IER register.

Figure 18-22. UART Interrupt Identification Register (IIR)

15–8	7–6	5–1	0
Reserved	FCR_MIRROR	IT_TYPE	IT_PENDING
R-0	R-0	R-0	R-1

LEGEND: R = Read only; -*n* = value after reset

Table 18-36. UART Interrupt Identification Register (IIR) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7-6	FCR_MIRROR	0-3h	Mirror the contents of FCR[0] on both bits.
5-1	IT_TYPE	0-1Fh	Seven possible interrupts in UART mode; other combinations never occur:
		0	Modem interrupt. Priority = 4.
		1h	THR interrupt. Priority = 3.
		2h	RHR interrupt. Priority = 2.
		3h	Receiver line status error. Priority = 1.
		4h-5h	Reserved
		6h	Rx timeout. Priority = 2.
		7h	Reserved
		8h	Xoff/special character. Priority = 5.
		9h-Fh	Reserved
		10h	$\overline{CTS}$, $\overline{RTS}$, $\overline{DSR}$ change state from active (low) to inactive (high). Priority = 6.
		11h-1Fh	Reserved
0	IT_PENDING		Interrupt pending.
		0	An interrupt is pending.
		1	No interrupt is pending.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.7 Interrupt Identification Register (IIR) - IrDA Mode

The IrDA interrupt identification register (IIR) is a read-only register that provides the source of the interrupt. The IrDA interrupt identification register (IIR) is shown in Figure 18-23 and described in Table 18-37.

NOTE: An interrupt source can be flagged only if enabled in the IER register.

Figure 18-23. IrDA Interrupt Identification Register (IIR)

15							8
Reserved							
R-0							
7	6	5	4	3	2	1	0
EOF_IT	LINE_STS_IT	TX_STATUS_IT	STS_FIFO_IT	RX_OE_IT	RX_FIFO_LAST_BYTE_IT	THR_IT	RHR_IT
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-37. IrDA Interrupt Identification Register (IIR) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7	EOF_IT	0	Received EOF interrupt inactive.
		1	Received EOF interrupt active.
6	LINE_STS_IT	0	Receiver line status interrupt inactive.
		1	Receiver line status interrupt active.
5	TX_STATUS_IT	0	TX status interrupt inactive.
		1	TX status interrupt active.
4	STS_FIFO_IT	0	Status FIFO trigger level interrupt inactive.
		1	Status FIFO trigger level interrupt active.
3	RX_OE_IT	0	RX overrun interrupt inactive.
		1	RX overrun interrupt active.
2	RX_FIFO_LAST_BYTE_IT	0	Last byte of frame in RX FIFO interrupt inactive.
		1	Last byte of frame in RX FIFO interrupt active.
1	THR_IT	0	THR interrupt inactive.
		1	THR interrupt active.
0	RHR_IT	0	RHR interrupt inactive.
		1	RHR interrupt active.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.8 Interrupt Identification Register (IIR) - CIR Mode

The CIR interrupt identification register (IIR) is a read-only register that provides the source of the interrupt. The CIR interrupt identification register (IIR) is shown in Figure 18-24 and described in Table 18-38.

NOTE: An interrupt source can be flagged only if enabled in the IER register.

Figure 18-24. CIR Interrupt Identification Register (IIR)

15							8
Reserved							
R-0							

7	6	5	4	3	2	1	0
Reserved		TXSTATUSIT	Reserved	RXOEIT	RXSTOPIT	THRIT	RHRIT
R-0		R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-38. CIR Interrupt Identification Register (IIR) Field Descriptions

Bit	Field	Value	Description
15-6	Reserved	0	Reserved.
5	TXSTATUSIT	0	TX status interrupt inactive
		1	TX status interrupt active
4	Reserved	0	Reserved.
3	RXOEIT	0	RX overrun interrupt inactive
		1	RX overrun interrupt active
2	RXSTOPIT	0	Receive stop interrupt is inactive
		1	Receive stop interrupt is active
1	THRIT	0	THR interrupt inactive
		1	THR interrupt active
0	RHRIT	0	RHR interrupt inactive
		1	RHR interrupt active

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.9 FIFO Control Register (FCR)

The FIFO Control Register (FCR) is shown in Figure 18-25 and described in Table 18-39.

Figure 18-25. FIFO Control Register (FCR)

15							8
Reserved							
R-0							
7	6	5	4	3	2	1	0
RX_FIFO_TRIG		TX_FIFO_TRIG		DMA_MODE	TX_FIFO_CLEAR	RX_FIFO_CLEAR	FIFO_EN
W-0		W-0		W-0	W-0	W-0	W-0

LEGEND: R = Read only; W = Write only; -*n* = value after reset

Table 18-39. FIFO Control Register (FCR) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7-6	RX_FIFO_TRIG	0-3h	Sets the trigger level for the RX FIFO: If SCR[7] = 0 and TLR[7:4] ≠ 0000, RX_FIFO_TRIG is not considered. If SCR[7] = 1, RX_FIFO_TRIG is 2 LSB of the trigger level (1 to 63 on 6 bits) with the granularity 1. If SCR[7] = 0 and TLR[7:4] = 0000:
		0	8 characters
		1h	16 characters
		2h	56 characters
		3h	60 characters
5-4	TX_FIFO_TRIG	0-3h	Can be written only if EFR[4] = 1. Sets the trigger level for the TX FIFO: If SCR[6] = 0 and TLR[3:0] ≠ 0000, TX_FIFO_TRIG is not considered. If SCR[6] = 1, TX_FIFO_TRIG is 2 LSB of the trigger level (1 to 63 on 6 bits) with a granularity of 1. If SCR[6] = 0 and TLR[3:0] = 0000:
		0	8 characters
		1h	16 characters
		2h	32 characters
		3h	56 characters
3	DMA_MODE		Can be changed only when the baud clock is not running (DLL and DLH cleared to 0). If SCR[0] = 0, this register is considered.
		0	DMA_MODE 0 (No DMA).
		1	DMA_MODE 1 (UART_NDMA_REQ[0] in TX, UART_NDMA_REQ[1] in RX).
2	TX_FIFO_CLEAR	0	No change.
		1	Clears the transmit FIFO and resets its counter logic to 0. Returns to 0 after clearing FIFO.
1	RX_FIFO_CLEAR	0	No change.
		1	Clears the receive FIFO and resets its counter logic to 0. Returns to 0 after clearing FIFO.
0	FIFO_EN		Can be changed only when the baud clock is not running (DLL and DLH cleared to 0).
		0	Disables the transmit and receive FIFOs. The transmit and receive holding registers are 1-byte FIFOs.
		1	Enables the transmit and receive FIFOs. The transmit and receive holding registers are 64-byte FIFOs.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.10 Line Control Register (LCR)

The line control register (LCR) is shown in Figure 18-26 and described in Table 18-40.

NOTE: As soon as LCR[6] is set to 1, the TX line is forced to 0 and remains in this state as long as LCR[6] = 1.

Figure 18-26. Line Control Register (LCR)

15						8
Reserved						
R-0						

7	6	5	4	3	2	1	0
DIV_EN	BREAK_EN	PARITY_TYPE2	PARITY_TYPE1	PARITY_EN	NB_STOP	CHAR_LENGTH	
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-40. Line Control Register (LCR) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7	DIV_EN		Divisor latch enable.
		0	Normal operating condition.
		1	Divisor latch enable. Allows access to DLL and DLH.
6	BREAK_EN		Break control bit.
			Note: When LCR[6] is set to 1, the TX line is forced to 0 and remains in this state as long as LCR[6] = 1.
		0	Normal operating condition.
		1	Forces the transmitter output to go low to alert the communication terminal.
5	PARITY_TYPE2		If LCR[3] = 1:
		0	If LCR[5] = 0, LCR[4] selects the forced parity format.
		1	If LCR[5] = 1 and LCR[4] = 0, the parity bit is forced to 1 in the transmitted and received data.
		1	If LCR[5] = 1 and LCR[4] = 1, the parity bit is forced to 0 in the transmitted and received data.
4	PARITY_TYPE1		If LCR[3] = 1:
		0	Odd parity is generated.
		1	Even parity is generated.
3	PARITY_EN		Parity bit.
		0	No parity.
		1	A parity bit is generated during transmission, and the receiver checks for received parity.
2	NB_STOP		Specifies the number of stop bits.
		0	1 stop bit (word length = 5, 6, 7, 8).
		1	1.5 stop bits (word length = 5) or 2 stop bits (word length = 6, 7, 8).
1-0	CHAR_LENGTH	0-3h	Specifies the word length to be transmitted or received.
		0	5 bits
		1h	6 bits
		2h	7 bits
		3h	8 bit

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.11 Modem Control Register (MCR)

Bits 3-0 control the interface with the modem, data set, or peripheral device that is emulating the modem. The modem control register (MCR) is shown in Figure 18-27 and described in Table 18-41.

Figure 18-27. Modem Control Register (MCR)

15							8
Reserved							
R-0							
7	6	5	4	3	2	1	0
Reserved	TCRTLR	XONEN	LOOPBACKEN	CDSTSCH	RISTSCH	RTS	DTR
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-41. Modem Control Register (MCR) Field Descriptions

Bit	Field	Value	Description
15-7	Reserved	0	Reserved.
6	TCRTLR		Can be written only when EFR[4] = 1.
		0	No action.
		1	Enables access to the TCR and TLR registers.
5	XONEN		Can be written only when EFR[4] = 1.
		0	Disable XON any function.
		1	Enable XON any function.
4	LOOPBACKEN		Loopback mode enable.
		0	Normal operating mode.
		1	Enable local loopback mode (internal). In this mode, the MCR[3:0] signals are looped back into MSR[7:4]. The transmit output is looped back to the receive input internally.
3	CDSTSCH	0	In loopback mode, forces $\overline{DCD}$ input high and IRQ outputs to INACTIVE state.
		1	In loopback mode, forces $\overline{DCD}$ input low and IRQ outputs to INACTIVE state.
2	RISTSCH	0	In loopback mode, forces $\overline{RI}$ input inactive (high).
		1	In loopback mode, forces $\overline{RI}$ input active (low).
1	RTS		In loopback mode, controls MSR[4]. If auto-RTS is enabled, the $\overline{RTS}$ output is controlled by hardware flow control.
		0	Force $\overline{RTS}$ output to inactive (high).
		1	Force $\overline{RTS}$ output to active (low).
0	DTR	0	Force $\overline{DTR}$ output (used in loopback mode) to inactive (high).
		1	Force $\overline{DTR}$ output (used in loopback mode) to active (low).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.12 Line Status Register (LSR) - UART Mode

When the UART line status register (LSR) is read, LSR[4:2] reflect the error bits (BI, FE, PE) of the character at the top of the RX FIFO (next character to be read). Therefore, reading the LSR and then reading the RHR identifies errors in a character. Reading RHR updates BI, FE, and PE. LSR [7] is set when there is an error anywhere in the RX FIFO and is cleared only when there are no more errors remaining in the RX FIFO. The UART line status register (LSR) is shown in Figure 18-28 and described in Table 18-42.

NOTE: Reading the LSR does not cause an increment of the RX FIFO read pointer. The RX FIFO read pointer is incremented by reading the RHR. Reading LSR clears OE if set.

Figure 18-28. UART Line Status Register (LSR)

15							8
Reserved							
R-0							
7	6	5	4	3	2	1	0
RXFIFOSTS	TXSRE	TXFIFOE	RXBI	RXFE	RXPE	RXOE	RXFIFOE
R-0	R-1	R-1	R-0	R-0	R-0	R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-42. UART Line Status Register (LSR) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7	RXFIFOSTS	0	Normal operation.
		1	At least one parity error, framing error, or break indication in the RX FIFO. Bit 7 is cleared when no errors are present in the RX FIFO.
6	TXSRE	0	Transmitter hold (TX FIFO) and shift registers are not empty.
		1	Transmitter hold (TX FIFO) and shift registers are empty.
5	TXFIFOE	0	Transmit hold register (TX FIFO) is not empty.
		1	Transmit hold register (TX FIFO) is empty. The transmission is not necessarily completed.
4	RXBI	0	No break condition.
		1	A break was detected while the data being read from the RX FIFO was being received (RX input was low for one character + 1 bit time frame).
3	RXFE	0	No framing error in data being read from RX FIFO.
		1	Framing error occurred in data being read from RX FIFO (received data did not have a valid stop bit).
2	RXPE	0	No parity error in data being read from RX FIFO.
		1	Parity error in data being read from RX FIFO.
1	RXOE	0	No overrun error.
		1	Overrun error occurred. Set when the character held in the receive shift register is not transferred to the RX FIFO. This case occurs only when receive FIFO is full.
0	RXFIFOE	0	No data in the receive FIFO.
		1	At least one data character in the RX FIFO.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.13 Line Status Register (LSR) - IrDA Mode

When the IrDA line status register (LSR) is read, LSR[4:2] reflect the error bits (FL, CRC, ABORT) of the frame at the top of the status FIFO (next frame status to be read). The IrDA line status register (LSR) is shown in Figure 18-29 and described in Table 18-43.

Figure 18-29. IrDA Line Status Register (LSR)

15							8
Reserved							
R-0							
7	6	5	4	3	2	1	0
THR_EMPTY	STS_FIFO_FULL	RX_LAST_BYTE	FRAME_TOO_LONG	ABORT	CRC	STS_FIFO_E	RX_FIFO_E
R-1	R-0	R-1	R-0	R-0	R-0	R-1	R-1

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-43. IrDA Line Status Register (LSR) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7	THR_EMPTY	0	Transmit holding register (TX FIFO) is not empty.
		1	Transmit hold register (TX FIFO) is empty. The transmission is not necessarily completed.
6	STS_FIFO_FULL	0	Status FIFO is not full.
		1	Status FIFO is full.
5	RX_LAST_BYTE	0	The RX FIFO (RHR) does not contain the last byte of the frame to be read.
		1	The RX FIFO (RHR) contains the last byte of the frame to be read. This bit is set to 1 only when the last byte of a frame is available to be read. It is used to determine the frame boundary. It is cleared on a single read of the LSR register.
4	FRAME_TOO_LONG	0	No frame-too-long error in frame.
		1	Frame-too-long error in the frame at the top of the status FIFO (next character to be read). This bit is set to 1 when a frame exceeding the maximum length (set by RXFLH and RXFLL registers) is received. When this error is detected, current frame reception is terminated. Reception is stopped until the next START flag is detected.
3	ABORT	0	No abort pattern error in frame.
		1	Abort pattern received. SIR and MIR: abort pattern. FIR: Illegal symbol.
2	CRC	0	No CRC error in frame.
		1	CRC error in the frame at the top of the status FIFO (next character to be read).
1	STS_FIFO_E	0	Status FIFO is not empty.
		1	Status FIFO is empty.
0	RX_FIFO_E	0	At least one data character in the RX FIFO.
		1	No data in the receive FIFO.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.14 Line Status Register (LSR) - CIR Mode

The CIR line status register (LSR) is shown in Figure 18-30 and described in Table 18-44.

Figure 18-30. CIR Line Status Register (LSR)

15				8
Reserved				
R-0				

7	6	5	4	1	0
THREMPTY	Reserved	RXSTOP	Reserved		RXFIFOE
R-1	R-0	R-0	R-0		R-1

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 18-44. CIR Line Status Register (LSR) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7	THREMPTY	0	Transmit holding register (TX FIFO) is not empty.
		1	Transmit hold register (TX FIFO) is empty. The transmission is not necessarily completed.
6	Reserved	0	Reserved.
5	RXSTOP		The RXSTOP is generated based on the value set in the BOF Length register (EBLR).
		0	Reception is on going or waiting for a new frame.
		1	Reception is completed. It is cleared on a single read of the LSR register.
4-1	Reserved	0	Reserved
0	RXFIFOE	0	At least one data character in the RX FIFO.
		1	No data in the receive FIFO.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.15 Modem Status Register (MSR)

The modem status register (MSR) provides information about the current state of the control lines from the modem, data set, or peripheral device to the Local Host. It also indicates when a control input from the modem changes state. The modem status register (MSR) is shown in Figure 18-31 and described in Table 18-45.

Figure 18-31. Modem Status Register (MSR)

15							8
Reserved							
R-0							
7	6	5	4	3	2	1	0
NCD_STS	NRI_STS	NDSR_STS	NCTS_STS	DCD_STS	RI_STS	DSR_STS	CTS_STS
R-unknown	R-unknown	R-unknown	R-unknown	R-0	R-0	R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-45. Modem Status Register (MSR) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7	NCD_STS		This bit is the complement of the $\overline{DCD}$ input. In loopback mode, it is equivalent to MCR[3].
6	NRI_STS		This bit is the complement of the $\overline{RI}$ input. In loopback mode, it is equivalent to MCR[2].
5	NDSR_STS		This bit is the complement of the $\overline{DSR}$ input. In loopback mode, it is equivalent to MCR[0].
4	NCTS_STS		This bit is the complement of the $\overline{CTS}$ input. In loopback mode, it is equivalent to MCR[1].
3	DCD_STS	0	No change.
		1	Indicates that $\overline{DCD}$ input (or MCR[3] in loopback mode) has changed. Cleared on a read.
2	RI_STS	0	No change.
		1	Indicates that $\overline{RI}$ input (or MCR[2] in loopback mode) changed state from low to high. Cleared on a read.
1	DSR_STS	0	No change.
		1	Indicates that $\overline{DSR}$ input (or MCR[0] in loopback mode) changed state. Cleared on a read.
0	CTS_STS	0	No change.
		1	Indicates that $\overline{CTS}$ input (or MCR[1] in loopback mode) changed state. Cleared on a read.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.16 Transmission Control Register (TCR)

The transmission control register (TCR) stores the receive FIFO threshold levels to start/stop transmission during hardware flow control. The transmission control register (TCR) is shown in Figure 18-32 and described in Table 18-46.

NOTE:

- Trigger levels from 0-60 bytes are available with a granularity of 4.
- Trigger level = 4 × [4-bit register value]
- You must ensure that TCR[3:0] > TCR[7:4], whenever auto-RTS or software flow control is enabled to avoid a misoperation of the device. In FIFO interrupt mode with flow control, you have to also ensure that the trigger level to HALT transmission is greater or equal to receive FIFO trigger level (either TLR[7:4] or FCR[7:6]); otherwise, FIFO operation stalls.
- In FIFO DMA mode with flow control, this concept does not exist because the DMA request is sent each time a byte is received.

Figure 18-32. Transmission Control Register (TCR)

15–8	7–4	3–0
Reserved	RXFIFOTRIGSTART	RXFIFOTRIGHALT
R-0	R/W-0	R/W-Fh

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-46. Transmission Control Register (TCR) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7-4	RXFIFOTRIGSTART	0-Fh	RX FIFO trigger level to RESTORE transmission (0 to 60).
3-0	RXFIFOTRIGHALT	0-Fh	RX FIFO trigger level to HALT transmission (0 to 60).

18.5.17 Scratchpad Register (SPR)

The scratchpad register (SPR) is a read/write register that does not control the module. It is a scratchpad register used to hold temporary data. The scratchpad register (SPR) is shown in Figure 18-33 and described in Table 18-47.

Figure 18-33. Scratchpad Register (SPR)

15–8	7–0
Reserved	SPR_WORD
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-47. Scratchpad Register (SPR) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7-0	SPR_WORD	0-FFh	Scratchpad register.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.18 Trigger Level Register (TLR)

This register stores the programmable transmit and receive FIFO trigger levels used for DMA and IRQ generation. The trigger level register (TLR) is shown in Figure 18-34 and described in Table 18-48.

Figure 18-34. Trigger Level Register (TLR)

15 – 8	7 – 4	3 – 0
Reserved	RX_FIFO_TRIG_ DMA	TX_FIFO_TRIG_ DMA
R-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-48. Trigger Level Register (TLR) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7-4	RX_FIFO_TRIG_ DMA	0-Fh	Receive FIFO trigger level. See Table 18-49.
3-0	TX_FIFO_TRIG_ DMA	0-Fh	Transmit FIFO trigger level. See Table 18-50.

Table 18-49. RX FIFO Trigger Level Setting Summary

SCR[7]	TLR[7:4]	Description
0	0	Defined by FCR[7:6] (either 8, 16, 56, 60 characters).
0	≠ 0000	Defined by TLR[7:4] (from 4 to 60 characters with a granularity of 4 characters).
1	any value	Defined by the concatenated value of TLR[7:4] and FCR[7:6] (from 1 to 63 characters with a granularity of 1 character). **Note:** the combination of TLR[7:4] = 0000 and FCR[7:6] = 00 (all zeros) is not supported (minimum of 1 character is required). All zeros results in unpredictable behavior.

Table 18-50. TX FIFO Trigger Level Setting Summary

SCR[6]	TLR[3:0]	Description
0	0	Defined by FCR[5:4] (either 8, 16, 32, 56 characters).
0	≠ 0000	Defined by TLR[3:0] (from 4 to 60 characters with a granularity of 4 characters).
1	any value	Defined by the concatenated value of TLR[3:0] and FCR[5:4] (from 1 to 63 characters with a granularity of 1 character). **Note:** the combination of TLR[3:0] = 0000 and FCR[5:4] = 00 (all zeros) is not supported (minimum of 1 character is required). All zeros results in unpredictable behavior.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.19 Mode Definition Register 1 (MDR1)

The mode of operation is programmed by writing to MDR1[2:0]; therefore, the mode definition register 1 (MDR1) must be programmed on startup after configuration of the configuration registers (DLL, DLH, and LCR). The value of MDR1[2:0] must not be changed again during normal operation. The mode definition register 1 (MDR1) is shown in Figure 18-35 and described in Table 18-51.

NOTE: If the module is disabled by setting the MODESELECT field to 7h, interrupt requests can still be generated unless disabled through the interrupt enable register (IER). In this case, UART mode interrupts are visible. Reading the interrupt identification register (IIR) shows the UART mode interrupt flags.

Figure 18-35. Mode Definition Register 1 (MDR1)

15–8					
Reserved					
R-0					

7	6	5	4	3	2–0
FRAMEENDMODE	SIPMODE	SCT	SETTXIR	IRSLEEP	MODESELECT
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-7h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-51. Mode Definition Register 1 (MDR1) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7	FRAMEENDMODE		IrDA mode only.
		0	Frame-length method.
		1	Set EOT bit method.
6	SIPMODE		MIR/FIR modes only.
		0	Manual SIP mode: SIP is generated with the control of ACREG[3].
		1	Automatic SIP mode: SIP is generated after each transmission.
5	SCT		Store and control the transmission.
		0	Starts the infrared transmission when a value is written to the THR register.
		1	Starts the infrared transmission with the control of ACREG[2]. **Note:** Before starting any transmission, there must be no reception ongoing.
4	SETTXIR		Used to configure the infrared transceiver.
		0	If MDR2[7] = 0, no action; if MDR2[7] = 1, TXIR pin output is forced low.
		1	TXIR pin output is forced high (not dependant of MDR2[7] value).
3	IRSLEEP		IrDA/CIR sleep mode.
		0	IrDA/CIR sleep mode disabled.
		1	IrDA/CIR sleep mode enabled.
2-0	MODESELECT	0-7h	UART/IrDA/CIR mode selection.
		0	UART 16× mode.
		1	SIR mode.
		2h	UART 16× auto-baud.
		3h	UART 13× mode.
		4h	MIR mode.
		5h	FIR mode.
		6h	CIR mode.
		7h	Disable (default state).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.20 Mode Definition Register 2 (MDR2)

The MDR2[0] bit describes the status of the TX status interrupt in IIR[5]. The IRTXUNDERRUN bit must be read after a TX status interrupt occurs. The MDR2[2:1] bits set the trigger level for the frame status FIFO (8 entries) and must be programmed before the mode is programmed in MDR1[2:0]. The mode definition register 2 (MDR2) is shown in Figure 18-36 and described in Table 18-52.

NOTE: The MDR2[6] bit gives the flexibility to invert the RX pin inside the UART module to ensure that the protocol at the input of the transceiver module has the same polarity at module level. By default, the RX pin is inverted because most of transceiver invert the IR receive pin.

Figure 18-36. Mode Definition Register 2 (MDR2)

15							8
Reserved							
R-0							

7	6	5	4	3	2	1	0
SETTXIRALT	IRRXINVERT	CIRPULSEMODE		UARTPULSE	STSFIFOTRIG		IRTXUNDERRUN
R/W-0	R/W-0	R/W-0		R/W-0	R/W-0		R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-52. Mode Definition Register 2 (MDR2) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7	SETTXIRALT		Provides alternate functionality for MDR1[4].
		0	Normal mode
		1	Alternate mode for SETTXIR
6	IRRXINVERT		Only for IR mode (IrDA and CIR). Invert RX pin in the module before the voting or sampling system logic of the infrared block. This does not affect the RX path in UART modem modes.
		0	Inversion is performed.
		1	No inversion is performed.
5-4	CIRPULSEMODE	0-3h	CIR pulse modulation definition. Defines high level of the pulse width associated with a digit:
		0	Pulse width of 3 from 12 cycles.
		1h	Pulse width of 4 from 12 cycles.
		2h	Pulse width of 5 from 12 cycles.
		3h	Pulse width of 6 from 12 cycles.
3	UARTPULSE		UART mode only. Used to allow pulse shaping in UART mode.
		0	Normal UART mode.
		1	UART mode with pulse shaping.
2-1	STSFIFOTRIG	0-3h	Only for IrDA mode. Frame status FIFO threshold select:
		0	1 entry
		1h	4 entries
		2h	7 entries
		3h	8 entries
0	IRTXUNDERRUN		IrDA transmission status interrupt. When the TX status interrupt (IIR[5]) occurs, the meaning of the interrupt is:
		0	The last bit of the frame was transmitted successfully without error.
		1	An underrun occurred. The last bit of the frame was transmitted but with an underrun error. The bit is reset to 0 when the RESUME register is read.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.21 Status FIFO Line Status Register (SFLSR)

Reading the status FIFO line status register (SFLSR) effectively reads frame status information from the status FIFO. This register does not physically exist. Reading this register increments the status FIFO read pointer (SFREGL and SFREGH must be read first). The status FIFO line status register (SFLSR) is shown in Figure 18-37 and described in Table 18-53.

NOTE: Top of RX FIFO = Next frame to be read from RX FIFO.

Figure 18-37. Status FIFO Line Status Register (SFLSR)

15–8	7–5	4	3	2	1	0
Reserved	Reserved	OE_ERROR	FRAME_TOO_LONG_ERROR	ABORT_DETECT	CRC_ERROR	Reserved
R-0	R-0	R-unknown	R-unknown	R-unknown	R-unknown	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-53. Status FIFO Line Status Register (SFLSR) Field Descriptions

Bit	Field	Value	Description
15-5	Reserved	0	Reserved.
4	OE_ERROR	0	No error
		1	Overrun error in RX FIFO when frame at top of RX FIFO was received.
3	FRAME_TOO_LONG_ERROR	0	No error
		1	Frame-length too long error in frame at top of RX FIFO.
2	ABORT_DETECT	0	No error
		1	Abort pattern detected in frame at top of RX FIFO.
1	CRC_ERROR	0	No error
		1	CRC error in frame at top of RX FIFO.
0	Reserved	0	Reserved.

18.5.22 RESUME Register

The RESUME register is used to clear internal flags, which halt transmission/reception when an underrun/overrun error occurs. Reading this register resumes the halted operation. This register does not physically exist and always reads as 00. The RESUME register is shown in Figure 18-38 and described in Table 18-54.

Figure 18-38. RESUME Register

15–8	7–0
Reserved	RESUME
R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-54. RESUME Register Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7-0	RESUME	0-FFh	Dummy read to restart the TX or RX.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.23 Status FIFO Register Low (SFREGL)

The frame lengths of received frames are written into the status FIFO. This information can be read by reading the status FIFO register low (SFREGL) and the status FIFO register high (SFREGH). These registers do not physically exist. The LSBs are read from SFREGL and the MSBs are read from SFREGH. Reading these registers does not alter the status FIFO read pointer. These registers must be read before the pointer is incremented by reading the SFLSR. The status FIFO register low (SFREGL) is shown in Figure 18-39 and described in Table 18-55.

Figure 18-39. Status FIFO Register Low (SFREGL)

15–8	7–0
Reserved	SFREGL
R-0	R-unknown

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-55. Status FIFO Register Low (SFREGL) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7-0	SFREGL	0-FFh	LSB part of the frame length.

18.5.24 Status FIFO Register High (SFREGH)

The frame lengths of received frames are written into the status FIFO. This information can be read by reading the status FIFO register low (SFREGL) and the status FIFO register high (SFREGH). These registers do not physically exist. The LSBs are read from SFREGL and the MSBs are read from SFREGH. Reading these registers does not alter the status FIFO read pointer. These registers must be read before the pointer is incremented by reading the SFLSR. The status FIFO register high (SFREGH) is shown in Figure 18-40 and described in Table 18-56.

Figure 18-40. Status FIFO Register High (SFREGH)

15–4	3–0
Reserved	SFREGH
R-0	R-unknown

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-56. Status FIFO Register High (SFREGH) Field Descriptions

Bit	Field	Value	Description
15-4	Reserved	0	Reserved.
3-0	SFREGH	0-Fh	MSB part of the frame length.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.25 BOF Control Register (BLR)

The BLR[6] bit is used to select whether C0h or FFh start patterns are to be used, when multiple start flags are required in SIR mode. If only one start flag is required, this is always C0h. If *n* start flags are required, either (n – 1) C0h or (n –1) FFh flags are sent, followed by a single C0h flag (immediately preceding the first data byte). The BOF control register (BLR) is shown in Figure 18-41 and described in Table 18-57.

Figure 18-41. BOF Control Register (BLR)

15–8	7	6	5–0
Reserved	STSFIFORESET	XBOFTYPE	Reserved
R-0	R/W-0	R/W-1	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-57. BOF Control Register (BLR) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7	STSFIFORESET		Status FIFO reset. This bit is self-clearing..
6	XBOFTYPE		SIR xBOF select.
		0	FFh start pattern is used.
		1	C0h start pattern is used.
0	Reserved	0	Reserved.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.26 Auxiliary Control Register (ACREG)

The auxiliary control register (ACREG) is shown in Figure 18-42 and described in Table 18-58.

NOTE:

- If transmit FIFO is not empty and MDR1[5] = 1, IrDA starts a new transfer with data of previous frame as soon as abort frame has been sent. Therefore, TX FIFO must be reset before sending an abort frame.
- It is recommended to disable TX FIFO underrun capability by masking corresponding underrun interrupt. When disabling underrun by setting ACREG[4] = 1, unknown data is sent over TX line.

Figure 18-42. Auxiliary Control Register (ACREG)

15							8
Reserved							
R-0							
7	6	5	4	3	2	1	0
PULSETYPE	SDMOD	DISIRRX	DISTXUNDERRUN	SENDSIP	SCTXEN	ABORTEN	EOTEN
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-58. Auxiliary Control Register (ACREG) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7	PULSETYPE		SIR pulse-width select:
		0	3/16 of baud-rate pulse width
		1	1.6 µs
6	SDMOD		Primary output used to configure transceivers. Connected to the SD/MODE input pin of IrDA transceivers.
		0	SD pin is set to high.
		1	SD pin is set to low.
5	DISIRRX		Disable RX input.
		0	Normal operation (RX input automatically disabled during transmit, but enabled outside of transmit operation).
		1	Disables RX input (permanent state; independent of transmit).
4	DISTXUNDERRUN		Disable TX underrun.
		0	Long stop bits cannot be transmitted. TX underrun is enabled.
		1	Long stop bits can be transmitted.
3	SENDSIP		MIR/FIR modes only. Send serial infrared interaction pulse (SIP).
			If this bit is set during an MIR/FIR transmission, the SIP is sent at the end of it. This bit is automatically cleared at the end of the SIP transmission.
		0	No action.
		1	Send SIP pulse.
2	SCTXEN		Store and control TX start. When MDR1[5] = 1 and the LH writes 1 to this bit, the TX state-machine starts frame transmission. This bit is self-clearing.
1	ABORTEN		Frame abort. The LH can intentionally abort transmission of a frame by writing 1 to this bit. Neither the end flag nor the CRC bits are appended to the frame.
0	EOTEN		EOT (end-of-transmission) bit. The LH writes 1 to this bit just before it writes the last byte to the TX FIFO in the set-EOT bit frame-closing method. This bit is automatically cleared when the LH writes to the THR (TX FIFO).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.27 Supplementary Control Register (SCR)

The supplementary control register (SCR) is shown in Figure 18-43 and described in Table 18-59.

NOTE: Bit 4 enables the wake-up interrupt, but this interrupt is not mapped into the IIR register. Therefore, when an interrupt occurs and there is no interrupt pending in the IIR register, the SSR[1] bit must be checked. To clear the wake-up interrupt, bit SCR[4] must be reset to 0.

Figure 18-43. Supplementary Control Register (SCR)

15							8
Reserved							
R-0							
7	6	5	4	3	2	1	0
RXTRIGGRANU1	TXTRIGGRANU1	DSRIT	RXCTSDSRWAKEUPENABLE	TXEMPTYCTLIT	DMAMODE2		DMAMODECTL
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		R/W-0

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 18-59. Supplementary Control Register (SCR) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7	RXTRIGGRANU1	0	Disables the granularity of 1 for trigger RX level.
		1	Enables the granularity of 1 for trigger RX level.
6	TXTRIGGRANU1	0	Disables the granularity of 1 for trigger TX level.
		1	Enables the granularity of 1 for trigger TX level.
5	DSRIT	0	Disables $\overline{\text{DSR}}$ interrupt.
		1	Enables $\overline{\text{DSR}}$ interrupt.
4	RXCTSDSRWAKEUPENABLE		RX CTS wake-up enable.
		0	Disables the WAKE UP interrupt and clears SSR[1].
		1	Waits for a falling edge of RX, $\overline{\text{CTS}}$, or $\overline{\text{DSR}}$ pins to generate an interrupt.
3	TXEMPTYCTLIT	0	Normal mode for THR interrupt.
		1	THR interrupt is generated when TX FIFO and TX shift register are empty.
2-1	DMAMODE2	0-3h	Specifies the DMA mode valid if SCR[0] = 1:
		0	DMA mode 0 (no DMA).
		1h	DMA mode 1 (UARTnDMAREQ[0] in TX, UARTnDMAREQ[1] in RX)
		2h	DMA mode 2 (UARTnDMAREQ[0] in RX)
		3h	DMA mode 3 (UARTnDMAREQ[0] in TX)
0	DMAMODECTL	0	The DMAMODE is set with FCR[3].
		1	The DMAMODE is set with SCR[2:1].

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.28 Supplementary Status Register (SSR)

The supplementary status register (SSR) is shown in Figure 18-44 and described in Table 18-60.

NOTE: Bit 1 is reset only when SCR[4] is reset to 0.

Figure 18-44. Supplementary Status Register (SSR)

15				8
Reserved				
R-0				

7	3	2	1	0
Reserved		DMACOUNTERRST	RXCTSDSRWAKEUPSTS	TXFIFOFULL
R-0		R/W-1	R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-60. Supplementary Status Register (SSR) Field Descriptions

Bit	Field	Value	Description
15-3	Reserved	0	Reserved.
2	DMACOUNTERRST	0	The DMA counter will not be reset, if the corresponding FIFO is reset (via FCR[1] or FCR[2]).
		1	The DMA counter will be reset, if the corresponding FIFO is reset (via FCR[1] or FCR[2]).
1	RXCTSDSRWAKEUPSTS		Pin falling edge detection: Reset only when SCR[4] is reset to 0.
		0	No falling-edge event on RX, $\overline{CTS}$, and $\overline{DSR}$.
		1	A falling edge occurred on RX, $\overline{CTS}$, or $\overline{DSR}$.
0	TXFIFOFULL	0	TX FIFO is not full.
		1	TX FIFO is full.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.29 BOF Length Register (EBLR)

In IrDA SIR operation, the BOF length register (EBLR) specifies the number of BOF + xBOFs to transmit. The value set into this register must consider the BOF character; therefore, to send only one BOF with no XBOF, this register must be set to 1. To send one BOF with *n* XBOFs, this register must be set to *n* + 1. Furthermore, the value 0 sends 1 BOF plus 255 XBOFs.

In IrDA MIR mode, the BOF length register (EBLR) specifies the number of additional start flags (MIR protocol mandates a minimum of 2 start flags).

In CIR mode, the BOF length register (EBLR) specifies the number of consecutive zeros to be received before generating the RXSTOP interrupt (IIR[2]). All the received zeros are stored in the RX FIFO. When the register is cleared to 0, this feature is deactivated and always in reception state, which is disabled by setting the ACREG[5] bit to 1.

The BOF length register (EBLR) is shown in Figure 18-45 and described in Table 18-61.

NOTE: If the RX_STOP interrupt occurs before a byte boundary, the remaining bits of the last byte are filled with zeros and then passed into the RX FIFO.

Figure 18-45. BOF Length Register (EBLR)

15–8	7–0
Reserved	EBLR
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-61. BOF Length Register (EBLR) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7-0	EBLR		IrDA mode: This register allows definition of up to 176 xBOFs, the maximum required by IrDA specification. CIR mode: This register specifies the number of consecutive zeros to be received before generating the RXSTOP interrupt (IIR[2]).
		0	Feature disabled.
		1h	Generate RXSTOP interrupt after receiving 1 zero bit.
		...	
		FFh	Generate RXSTOP interrupt after receiving 255 zero bits.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.30 Module Version Register (MVR)

The reset value is fixed by hardware and corresponds to the RTL revision of this module. A reset has no effect on the value returned. The module version register (MVR) is shown in Figure 18-46 and described in Table 18-62.

Figure 18-46. Module Version Register (MVR)

15–8	7–4	3–0
Reserved	MAJORREV	MINORREV
R-0	R-unknown	R-unknown

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-62. Module Version Register (MVR) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7-4	MAJORREV	0-Fh	Major revision number of the module.
3-0	MINORREV	0-Fh	Minor revision number of the module.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.31 System Configuration Register (SYSC)

The AUTOIDLE bit controls a power-saving technique to reduce the logic power consumption of the module interface; that is, when the feature is enabled, the interface clock is gated off until the module interface is accessed. When the SOFTRESET bit is set high, it causes a full device reset. The system configuration register (SYSC) is shown in Figure 18-47 and described in Table 18-63.

Figure 18-47. System Configuration Register (SYSC)

15–5	4–3	2	1	0
Reserved	IDLEMODE	ENAWAKEUP	SOFTRESET	AUTOIDLE
R-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-63. System Configuration Register (SYSC) Field Descriptions

Bit	Field	Value	Description
15-5	Reserved	0	Reserved.
4-3	IDLEMODE	0-3h	Power management req/ack control.
		0	Force idle: Idle request is acknowledged unconditionally.
		1h	No-idle: Idle request is never acknowledged.
		2h	Smart idle: Acknowledgement to an idle request is given based in the internal activity of the module.
		3h	Smart idle Wakeup: Acknowledgement to an idle request is given based in the internal activity of the module. The module is allowed to generate wakeup request.
2	ENAWAKEUP		Wakeup control.
		0	Wakeup is disabled.
		1	Wakeup capability is enabled.
1	SOFTRESET		Software reset. Set this bit to 1 to trigger a module reset. This bit is automatically reset by the hardware. Read returns 0.
		0	Normal mode.
		1	Module is reset.
0	AUTOIDLE		Internal interface clock-gating strategy.
		0	Clock is running.
		1	Automatic interface clock-gating strategy is applied based on interface activity.

18.5.32 System Status Register (SYSS)

The system status register (SYSS) is shown in Figure 18-48 and described in Table 18-64.

Figure 18-48. System Status Register (SYSS)

15–1	0
Reserved	RESETDONE
R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-64. System Status Register (SYSS) Field Descriptions

Bit	Field	Value	Description
15-1	Reserved	0	Reserved.
0	RESETDONE		Internal reset monitoring.
		0	Internal module reset is ongoing.
		1	Reset complete.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.33 Wake-Up Enable Register (WER)

The wake-up enable register (WER) is used to mask and unmask a UART event that subsequently notifies the system. An event is any activity in the logic that can cause an interrupt and/or an activity that requires the system to wake up. Even if wakeup is disabled for certain events, if these events are also an interrupt to the UART, the UART still registers the interrupt as such. The wake-up enable register (WER) is shown in Figure 18-49 and described in Table 18-65.

Figure 18-49. Wake-Up Enable Register (WER)

15							8
Reserved							
R-0							
7	6	5	4	3	2	1	0
TXWAKEUPEN	RLS_INTERRUPT	RHR_INTERRUPT	RX_ ACTIVITY	DCD_ACTIVITY	RI_ ACTIVITY	DSR_ACTIVITY	CTS_ACTIVITY
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-65. Wake-Up Enable Register (WER) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7	TXWAKEUPEN		Wake-up interrupt.
		0	Event is not allowed to wake up the system.
		1	Event can wake up the system: Event can be: THRIT or TXDMA request and/or TXSATUSIT.
6	RLS_ INTERRUPT		Receiver line status interrupt.
		0	Event is not allowed to wake up the system.
		1	Event can wake up the system.
5	RHR_ INTERRUPT		RHR interrupt.
		0	Event is not allowed to wake up the system.
		1	Event can wake up the system.
4	RX_ ACTIVITY	0	Event is not allowed to wake up the system.
		1	Event can wake up the system.
3	DCD_ACTIVITY	0	Event is not allowed to wake up the system.
		1	Event can wake up the system.
2	RI_ ACTIVITY	0	Event is not allowed to wake up the system.
		1	Event can wake up the system.
1	DSR_ACTIVITY	0	Event is not allowed to wake up the system.
		1	Event can wake up the system.
0	CTS_ ACTIVITY	0	Event is not allowed to wake up the system.
		1	Event can wake up the system.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.34 Carrier Frequency Prescaler Register (CFPS)

Since the consumer IR (CIR) works at modulation rates of 30–56.8 kHz, the 48 MHz clock must be prescaled before the clock can drive the IR logic. The carrier frequency prescaler register (CFPS) sets the divisor rate to give a range to accommodate the remote control requirements in BAUD multiples of 12×. The value of the CFPS at reset is 105 decimal (69h), which equates to a 38.1 kHz output from starting conditions. The 48 MHz carrier is prescaled by the CFPS that is then divided by the 12× BAUD multiple. The carrier frequency prescaler register (CFPS) is shown in Figure 18-50 and described in Table 18-66.

Figure 18-50. Carrier Frequency Prescaler Register (CFPS)

15–8	7–0
Reserved	CFPS
R-0	R/W-69h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-66. Carrier Frequency Prescaler Register (CFPS) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7-0	CFPS	0-FFh	System clock frequency prescaler at (12× multiple). CFPS = 0 is not supported. Examples for CFPS values:

Target Frequency (kHz)	CFPS (decimal)	Actual Frequency (kHz)
30	133	30.08
32.75	122	32.79
36	111	36.04
36.7	109	36.69
38	105	38.1
40	100	40
56.8	70	57.14

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.35 Divisor Latches Low Register (DLL)

The divisor latches low register (DLL) with the DLH register stores the 14-bit divisor for generation of the baud clock in the baud rate generator. DLH stores the most-significant part of the divisor, DLL stores the least-significant part of the divisor. The DLL register is shown in Figure 18-51 and described in Table 18-67.

NOTE: DLL and DLH can be written to only before sleep mode is enabled (before IER[4] is set).

Figure 18-51. Divisor Latches Low Register (DLL)

15 – 8	7 – 0
Reserved	CLOCK_LSB
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-67. Divisor Latches Low Register (DLL) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7-0	CLOCK_LSB	0-FFh	Divisor latches low. Stores the 8 LSB divisor value.

18.5.36 Divisor Latches High Register (DLH)

The divisor latches high register (DLH) with the DLL register stores the 14-bit divisor for generation of the baud clock in the baud rate generator. DLH stores the most-significant part of the divisor, DLL stores the least-significant part of the divisor. The DLH register is shown in Figure 18-52 and described in Table 18-68.

NOTE: DLL and DLH can be written to only before sleep mode is enabled (before IER[4] is set).

Figure 18-52. Divisor Latches High Register (DLH)

15 – 6	5 – 0
Reserved	CLOCK_MSB
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-68. Divisor Latches High Register (DLH) Field Descriptions

Bit	Field	Value	Description
15-6	Reserved	0	Reserved.
5-0	CLOCK_MSB	0-3Fh	Divisor latches high. Stores the 6 MSB divisor value.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.37 Enhanced Feature Register (EFR)

The enhanced feature register (EFR) enables or disables enhanced features. Most enhanced functions apply only to UART modes, but EFR[4] enables write accesses to FCR[5:4], the TX trigger level, which is also used in IrDA modes. The enhanced feature register (EFR) is shown in Figure 18-53 and described in Table 18-69.

Figure 18-53. Enhanced Feature Register (EFR)

15					8
Reserved					
R-0					

7	6	5	4	3	0
AUTOCTSEN	AUTORTSEN	SPECIALCHARDETECT	ENHANCEDEN	SWFLOWCONTROL	
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-69. Enhanced Feature Register (EFR) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7	AUTOCTSEN		Auto-CTS enable bit (UART mode only).
		0	Normal operation.
		1	Auto-CTS flow control is enabled; transmission is halted when the $\overline{CTS}$ pin is high (inactive).
6	AUTORTSEN		Auto-RTS enable bit (UART mode only).
		0	Normal operation.
		1	Auto-RTS flow control is enabled; $\overline{RTS}$ pin goes high (inactive) when the receiver FIFO HALT trigger level, TCR[3:0], is reached and goes low (active) when the receiver FIFO RESTORE transmission trigger level is reached.
5	SPECIALCHARDETECT		Special character detect (UART mode only).
		0	Normal operation.
		1	Special character detect enable. Received data is compared with XOFF2 data. If a match occurs, the received data is transferred to RX FIFO and the IIR[4] bit is set to 1 to indicate that a special character was detected.
4	ENHANCEDEN		Enhanced functions write enable bit.
		0	Disables writing to IER[7:4], FCR[5:4], and MCR[7:5].
		1	Enables writing to IER[7:4], FCR[5:4], and MCR[7:5].
3-0	SWFLOWCONTROL	0-Fh	Combinations of software flow control can be selected by programming this bit. XON1 and XON2 should be set to different values if the software flow control is enabled. See Table 18-70. In IrDA mode, EFR[1:0] selects the IR address to check. See *IR Address Checking*.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 18-70. EFR[3:0] Software Flow Control Options

EFR[3:0]				
Bit 3	Bit 2	Bit 1	Bit 0	TX/RX Software Flow Control
0	0	X	X	No transmit flow control
1	0	X	X	Transmit XON1, XOFF1
0	1	X	X	Transmit XON2, XOFF2
1	1	X	X	Transmit XON1, XON2: XOFF1, XOFF2[(1)]
X	X	0	0	No receive flow control
X	X	1	0	Receiver compares XON1, XOFF1
X	X	0	1	Receiver compares XON2, XOFF2
X	X	1	1	Receiver compares XON1, XON2: XOFF1, XOFF2[(1)]

(1) The XON1 and XON2 characters or the XOFF1 and XOFF2 characters must be transmitted/received sequentially with XON1/XOFF1 followed by XON2/XOFF2.

18.5.38 XON1/ADDR1 Register

In UART mode, XON1 character; in IrDA mode, ADDR1 address 1. The XON1/ADDR1 register is shown in Figure 18-54 and described in Table 18-71.

Figure 18-54. XON1/ADDR1 Register

15–8	7–0
Reserved	XONWORD1
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-71. XON1/ADDR1 Register Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7-0	XONWORD1	0-FFh	Stores the 8-bit XON1 character in UART modes and ADDR1 address 1 in IrDA modes.

18.5.39 XON2/ADDR2 Register

In UART mode, XON2 character; in IrDA mode, ADDR2 address 2. The XON2/ADDR2 register is shown in Figure 18-55 and described in Table 18-72.

Figure 18-55. XON2/ADDR2 Register

15–8	7–0
Reserved	XONWORD2
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-72. XON2/ADDR2 Register Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
5	XONWORD2	0-FFh	Stores the 8-bit XON2 character in UART modes and ADDR2 address 2 in IrDA modes.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.40 XOFF1 Register

In UART mode, XOFF1 character. The XOFF1 register is shown in Figure 18-56 and described in Table 18-73.

Figure 18-56. XOFF1 Register

15 – 8	7 – 0
Reserved	XOFFWORD1
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-73. XOFF1 Register Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7-0	XOFFWORD1	0	Stores the 8-bit XOFF1 character in UART modes.

18.5.41 XOFF2 Register

In UART mode, XOFF2 character. The XOFF2 register is shown in Figure 18-57 and described in Table 18-74.

Figure 18-57. XOFF2 Register

15 – 8	7 – 0
Reserved	XOFFWORD2
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-74. XOFF2 Register Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7-0	XOFFWORD2	0-FFh	Stores the 8-bit XOFF2 character in UART modes.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.42 Transmit Frame Length Low Register (TXFLL)

The transmit frame length low register (TXFLL) and the TXFLH register hold the 13-bit transmit frame length (expressed in bytes). TXFLL holds the LSBs and TXFLH holds the MSBs. The frame length value is used if the frame length method of frame closing is used. The transmit frame length low register (TXFLL) is shown in Figure 18-58 and described in Table 18-75.

Figure 18-58. Transmit Frame Length Low Register (TXFLL)

15–8	7–0
Reserved	TXFLL
R-0	W-0

LEGEND: R/W = Read/Write; W = Write only; -*n* = value after reset

Table 18-75. Transmit Frame Length Low Register (TXFLL) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7-0	TXFLL	0-FFh	LSB register used to specify the frame length.

18.5.43 Transmit Frame Length High Register (TXFLH)

The transmit frame length high register (TXFLH) and the TXFLL register hold the 13-bit transmit frame length (expressed in bytes). TXFLL holds the LSBs and TXFLH holds the MSBs. The frame length value is used if the frame length method of frame closing is used. The transmit frame length high register (TXFLH) is shown in Figure 18-59 and described in Table 18-76.

Figure 18-59. Transmit Frame Length High Register (TXFLH)

15–5	4–0
Reserved	TXFLH
R-0	W-0

LEGEND: R/W = Read/Write; W = Write only; -*n* = value after reset

Table 18-76. Transmit Frame Length High Register (TXFLH) Field Descriptions

Bit	Field	Value	Description
15-5	Reserved	0	Reserved.
4-0	TXFLH	0-1Fh	MSB register used to specify the frame length.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.44 Received Frame Length Low Register (RXFLL)

The received frame length low register (RXFLL) and the RXFLH register hold the 12-bit receive maximum frame length. RXFLL holds the LSBs and RXFLH holds the MSBs. If the intended maximum receive frame length is *n* bytes, program RXFLL and RXFLH to be *n* + 3 in SIR or MIR modes and *n* + 6 in FIR mode (+3 and +6 are the result of frame format with CRC and stop flag; two bytes are associated with the FIR stop flag). The received frame length low register (RXFLL) is shown in Figure 18-60 and described in Table 18-77.

Figure 18-60. Received Frame Length Low Register (RXFLL)

15 – 8	7 – 0
Reserved	RXFLL
R-0	W-0

LEGEND: R/W = Read/Write; W = Write only; -*n* = value after reset

Table 18-77. Received Frame Length Low Register (RXFLL) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7-0	RXFLL	0-FFh	LSB register used to specify the frame length in reception.

18.5.45 Received Frame Length High Register (RXFLH)

The received frame length high register (RXFLH) and the RXFLL register hold the 12-bit receive maximum frame length. RXFLL holds the LSBs and RXFLH holds the MSBs. If the intended maximum receive frame length is *n* bytes, program RXFLL and RXFLH to be *n* + 3 in SIR or MIR modes and *n* + 6 in FIR mode (+3 and +6 are the result of frame format with CRC and stop flag; two bytes are associated with the FIR stop flag). The received frame length high register (RXFLH) is shown in Figure 18-61 and described in Table 18-78.

Figure 18-61. Received Frame Length High Register (RXFLH)

15 – 4	3 – 0
Reserved	RXFLH
R-0	W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-78. Received Frame Length High Register (RXFLH) Field Descriptions

Bit	Field	Value	Description
15-4	Reserved	0	Reserved.
3-0	RXFLH	0-Fh	MSB register used to specify the frame length in reception.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

18.5.46 UART Autobauding Status Register (UASR)

The UART autobauding status register (UASR) returns the speed, the number of bits by characters, and the type of parity in UART autobauding mode. In autobauding mode, the input frequency of the UART modem must be fixed to 48 MHz. Any other module clock frequency results in incorrect baud rate recognition. The UART autobauding status register (UASR) is shown in Figure 18-62 and described in Table 18-79.

NOTE:

- This register is used to set up transmission according to characteristics of previous reception, instead of LCR, DLL, and DLH registers when UART is in autobauding mode.
- To reset the autobauding hardware (to start a new "AT" detection) or to set the UART in standard mode (no autobaud), MDR1[2:0] must be set to 7h (reset state), then set to 2h (UART in autobaud mode) or cleared to 0 (UART in standard mode).

Usage limitation:

- Only 7 and 8 bits character (5 and 6 bits not supported).
- 7 bits character with space parity not supported.
- Baud rate between 1200 and 115 200 bp/s (10 possibilities).

Figure 18-62. UART Autobauding Status Register (UASR)

15–8	7–6	5	4–0
Reserved	PARITYTYPE	BITBYCHAR	SPEED
R-0	R-0	R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 18-79. UART Autobauding Status Register (UASR) Field Descriptions

Bit	Field	Value	Description
15-8	Reserved	0	Reserved.
7-6	PARITYTYPE	0-3h	Type of the parity in UART autobauding mode.
		0	No parity identified
		1h	Parity space
		2h	Even parity
		3h	Odd parity
5	BITBYCHAR		Number of bits by characters.
		0	7-bit character identified.
		1	8-bit character identified.
4-0	SPEED	0-1Fh	Speed
		0	No speed identified.
		1h	115 200 baud
		2h	57 600 baud
		3h	38 400 baud
		4h	28 800 baud
		5h	19 200 baud
		6h	14 400 baud
		7h	9600 baud
		8h	4800 baud
		9h	2400 baud
		Ah	1200 baud
		Bh-1Fh	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Chapter 19

SPRUH73–July 2011

Timer

This chapter describes the timers for the device.

Topic **Page**

19.1 DMTimer **2500**
19.2 DMTimer 1ms **2527**
19.3 RTC_SS **2556**
19.4 WATCHDOG **2585**

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.1 DMTimer

19.1.1 Introduction

19.1.1.1 Overview

The timer module contains a free running upward counter with auto reload capability on overflow. The timer counter can be read and written in real-time (while counting). The timer module includes compare logic to allow an interrupt event on a programmable counter matching value.

A dedicated output signal can be pulsed or toggled on overflow and match event. This output offers a timing stamp trigger signal or PWM (pulse-width modulation) signal sources. A dedicated output signal can be used for general purpose PORGPOCFG (directly driven by bit 14 of the TCLR register). A dedicated input signal is used to trigger automatic timer counter capture and interrupt event, on programmable input signal transition type. A programmable clock divider (prescaler) allows reduction of the timer input clock frequency. All internal timer interrupt sources are merged in one module interrupt line and one wake-up line. Each internal interrupt sources can be independently enabled/disabled.

This module is controllable through the OCP peripheral bus.

As two clock domains are managed inside this module, resynchronization is done by special logic between the OCP clock domain and the Timer clock domain. At reset, synchronization logic allows utilization of all ratios between the OCP clock and the Timer clock. A drawback of this mode is that full-resynchronization path is used with access latency performance impact in terms of OCP clock cycles. In order to improve module access latency, and under restricted conditions on clocks ratios, write-posted mode can be used by setting the POSTED bit of the System Control Register (TSCR). Under this mode, write posted mode is enabled, meaning that OCP write command is granted before write process complete in the timer clock domain. This mode allows software to do concurrent writes on Dual Mode timer registers and to observe write process completion (synchronization) at the software level by reading independent write posted status bits in the Write Posted Status Register (TWPS).

19.1.1.2 Features

The timer consists of the following features:

- Counter timer with compare and capture modes
- Auto-reload mode
- Start-stop mode
- Programmable divider clock source
- 16-32 bit addressing
- “On the fly” read/write registers
- Interrupts generated on overflow, compare and capture
- Interrupt enable
- Wake-up enable
- Write posted mode
- Dedicated input trigger for capture mode and dedicated output trigger/PWM signal
- Dedicated output signal for general purpose use PORGPOCFG
- OCP interface compatible

The Timer resolution and interrupt period are dependent on the selected input clock and clock prescale value. Example resolutions for common clock values are shown in Table 19-1.

Table 19-1. Timer Resolution and Maximum Range

Clock	Prescaler	Resolution	Interrupt Period Range
32.768 KHz	1 (min)	31.25 us	31.25 us to ~36h 35m
	256 (max)	8 ms	8 ms to ~391d 22h 48m

Copyright © 2011, Texas Instruments Incorporated

Table 19-1. Timer Resolution and Maximum Range (continued)

Clock	Prescaler	Resolution	Interrupt Period Range
27 MHz	1 (min)	~37 ns	~37 ns to ~159s
	256 (max)	~9.48 us	~19.7 us to ~11h 18m
38.4 MHz	1 (min)	~26 ns	~26 ns to ~112 s
	256 (max)	~6.7 us	~13.3 us to ~7h 5m

19.1.1.3 Functional Block Diagram

Figure 19-1 shows a block diagram of the timer.

Figure 19-1. Timer Block Diagram

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.1.2 Integration

19.1.2.1 Timer Connectivity Attributes

Table 19-2. Timer[2–7] Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_L4LS_GCLK (OCP) Functional Clocks: PD_PER_TIMER2_GCLK (Timer 2) PD_PER_TIMER3_GCLK (Timer 3) PD_PER_TIMER4_GCLK (Timer 4) PD_PER_TIMER5_GCLK (Timer 5) PD_PER_TIMER6_GCLK (Timer 6) PD_PER_TIMER7_GCLK (Timer 7)
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	1 per timer module to MPU Subsystem Also to TSC_ADC event capture mux for Timer 4 – Timer7
DMA Requests	None
Physical Address	L4 Peripheral slave port

19.1.2.2 Timer Clock and Reset Management

Each DMTimer[2–7] functional clock is selected within the PRCM using the associated CLKSEL_TIMERx_CLK register from 3 possible sources:

- The 24 MHz (typ) system clock (CLK_M_OSC)
- The PER PLL generated 32.768 KHz clock (CLK_32KHZ)
- The TCLKIN external timer input clock.

19.1.2.3 Timer Clock Signals

Table 19-3. Timer Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
Timer[2–7] Clock Signals			
PICLKOCP Interface clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_l4ls_gclk from PRCM
PICLKTIMER Functional clock	25 MHz*	CLK_M_OSC CLK_32KHZ (PER_CLKOUTM2 / 5859.375) TCLKIN	pd_per_timer2_gclk pd_per_timer3_gclk pd_per_timer4_gclk pd_per_timer5_gclk pd_per_timer6_gclk pd_per_timer7_gclk from PRCM

19.1.2.4 Timer Pin List

The timer PIEVENTCAPT input and PORTIMERPWM output signals are muxed onto a single TIMER I/O pad. The pad direction (and hence the pin function) are controlled from within the DMTimer module using the PORGPOCFG signal as an output enable.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 19-4. Timer Pin List

Pin	Type	Description
TCLKIN	I	External timer clock source
TIMER4	I/O	Timer 4 trigger input or PWM output
TIMER5	I/O	Timer 5 trigger input or PWM output
TIMER6	I/O	Timer 6 trigger input or PWM output
TIMER7	I/O	Timer 7 trigger input or PWM output

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.1.3 Functional Description

19.1.3.1 Functional Description

The general-purpose timer is an upward counter. It supports 3 functional modes:

- Timer mode
- Capture mode
- Compare mode

By default, after core reset, the capture and compare modes are disabled.

19.1.3.1.1 Timer Mode Functionality

The timer is an upward counter that can be started and stopped at any time through the Timer Control Register (TCLR ST bit). The Timer Counter Register (TCRR) can be loaded when stopped or on the fly (while counting). TCRR can be loaded directly by a TCRR Write access with the new timer value. TCRR can also be loaded with the value held in the Timer Load Register TLDR by a trigger register (TTGR) Write access. The TCRR loading is done regardless the TTGR written value. The timer counter register TCRR value can be read when stopped or captured on the fly by a TCRR Read access. The timer is stopped and the counter value is cleared to “0” when the module’s reset is asserted. The timer is maintained in stop after reset is released. When the timer is stopped TCRR is frozen and it can be restarted from the frozen value unless TCRR has been reloaded with a new value.

In the one shot mode (TCLR AR bit = 0), the counter is stopped after counting overflow (counter value remains at zero).

When the auto-reload mode is enabled (TCLR AR bit = 1), the TCRR is reloaded with the Timer Load Register (TLDR) value after a counting overflow.

It is not recommended to put the overflow value (FFFF FFFFh) in TLDR because it can lead to undesired results.

An interrupt can be issued on overflow if the overflow interrupt enable bit is set in the timer Interrupt Enable Register (IRQENABLE_SET OVF_IT_FLAG bit = 1). A dedicated output pin (PORTIMERPWM) is programmed through TCLR (TRG and PT bits) to generate one positive pulse (prescaler duration) or to invert the current value (toggle mode) when an overflow occurs.

Figure 19-2. TCRR Timing Value

19.1.3.1.2 Capture Mode Functionality

The timer value in TCRR can be captured and saved in TCAR1 or TCAR2 function of the mode selected in TCLR through the field CAPT_MODE when a transition is detected on the module input pin (PIEVENTCAPT). The edge detection circuitry monitors transitions on the input pin (PIEVENTCAPT).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Rising transition, falling transition or both can be selected in TCLR (TCM bit) to trig the timer counter capture. The module sets the IRQSTATUS (TCAR_IT_FLAG bit) when an active transition is detected and at the same time the counter value TCRR is stored in one of the timer capture registers TCAR1 or TCAR2 as follows:

- If TCLR's CAPT_MODE field is 0 then, on the first enabled capture event, the value of the counter register is saved in TCAR1 register and all the next events are ignored (no update on TCAR1 and no interrupt triggering) until the detection logic is reset or the interrupt status register is cleared on TCAR's position writing a 1 in it.
- If TCLR's CAPT_MODE field is 1 then, on the first enabled captured event, the counter value is saved in TCAR1 register and, on the second enabled capture event, the value of the counter register is saved in TCAR2 register. All the other events are ignored (no update on TCAR1/2 and no interrupt triggering) until the detection logic is reset or the interrupt status register is cleared on TCAR's position writing a 1 in it. This mechanism is useful for period calculation of a clock if that clock is connected to the PIEVENTCAPT input pin.

The edge detection logic is reset (a new capture is enabled) when the active capture interrupt is served - TCAR_IT_FLAG bit of IRQSTATUS (previously 1) is cleared. The timer functional clock (input to prescaler) is used to sample the input pin (PIEVENTCAPT). Input negative or positive pulse can be detected when pulse time is above functional clock period. An interrupt can be issued on transition detection if the capture interrupt enable bit is set in the Timer Interrupt Enable Register IRQENABLE_SET (TCAR_IT_FLAG bit).

In Figure 19-3, the TCM value is 01 and CAPT_MODE is 0 - only rising edge of the PIEVENTCAPT will trigger a capture in TCAR and only TCAR1 will update.

In Figure 19-4, the TCM value is 01 and CAPT_MODE is 1 - only rising edge of the PIEVENTCAPT will trigger a capture in TCAR1 on first enabled event and TCAR2 will update on the second enabled event.

Figure 19-3. Capture Wave Example for CAPT_MODE = 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 19-4. Capture Wave Example for CAPT_MODE = 1

19.1.3.1.3 Compare Mode Functionality

When Compare Enable TCLR (CE bit) is set to 1, the timer value (TCRR) is permanently compared to the value held in timer match register (TMAR). TMAR value can be loaded at any time (timer counting or stop). When the TCRR and the TMAR values match, an interrupt can be issued if the IRQENABLE_SET (MAT_EN_FLAG bit) is set. The right programming way is to write a compare value in TMAR register before setting TCLR (CE bit) to avoid any unwanted interrupt due to a reset value matching effect.

The dedicated output pin (PORTIMERPWM) can be programmed through TCLR (TRG and PT bits) to generate one positive pulse (TIMER clock duration) or to invert the current value (toggle mode) when an overflow and a match occur.

19.1.3.1.4 Prescaler Functionality

A prescaler counter can be used to divide the timer counter input clock frequency. The prescaler is enabled when TCLR bit 5 is set (PRE). The 2n division ratio value (PTV) can be configured in the TCLR register. The prescaler counter is reset when the timer counter is stopped or reloaded on the fly.

Table 19-5. Prescaler Functionality

Contexts	Prescaler Counter	Timer Counter
Overflow (when Auto-reload on)	Reset	TLDR
TCRR Write	Reset	TCRR
TTGR Write	Reset	TLDR
Stop	Reset	Frozen

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.1.3.1.5 Pulse-Width Modulation

The timer can be configured to provide a programmable pulse-width modulation (PORTIMERPWM) output. The PORTIMERPWM output pin can be configured to toggle on specified event. TCLR (TRG bits) determines on which register value the PORTIMERPWM pin toggles. Either overflow or match can be used to toggle the PORTIMERPWM pin, when a compare condition occurs.

In case of overflow and match mode, the match event will be ignored from the moment the mode was set-up until the first overflow event occurs (see Figure 19-4).

The TCLR (SCPWM bit) can be programmed to set or clear the PORTIMERPWM output signal while the counter is stopped or the triggering is off only. This allows fixing a deterministic state of the output pin when modulation is stopped. The modulation is synchronously stopped when TRG bit is cleared and overflow occurred.

In the following timing diagram, the internal overflow pulse is set each time (FFFF FFFFFh – TLDR + 1) value is reached, and the internal match pulse is set when the counter reaches TMAR register value. According to TCLR (TRG and PT bits) programming value, the timer provides pulse or PWM on the output pin (PORTIMERPWM).

The TLDR and TMAR registers must keep values smaller than the overflow value (FFFF FFFFh) with at least 2 units. In case the PWM trigger events are both overflow and match, the difference between the values kept in TMAR register and the value in TLDR must be at least 2 units. When match event is used the compare mode TCLR (CE) must be set.

In Figure 19-5, TCLR (SCPWM bit) is cleared to 0. In Figure 19-6, TCLR (SCPWM bit) is set to 1.

Figure 19-5. Timing Diagram of Pulse-Width Modulation with SCPWM = 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 19-6. Timing Diagram of Pulse-Width Modulation with SCPWM = 1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.1.3.1.6 Timer Counting Rate

The timer counter is composed of a prescaler stage and a timer counter. Prescaler stage is clocked with the timer clock and acts as a clock divider for the timer counter stage. The ratio can be managed by accessing the ratio definition field of the control register (PTV and PRE of TCLR). See Table 19-6.

The timer rate is defined by:

- The value of the prescaler fields (PRE and PTV of TCLR register)
- The value loaded into the Timer Load Register (TLDR).

Table 19-6. Prescaler Clock Ratios Value

PRE	PTV	Divisor (PS)
0	X	1
1	0	2
1	1	4
1	2	8
1	3	16
1	4	32
1	5	64
1	6	128
1	7	256

The timer rate equation is as follows:

(FFFF FFFFh – TLDR + 1) × timer Clock period × Clock Divider (PS)

With timer Clock period = 1/ timer Clock frequency and PS = 2(PTV + 1).

As an example, if we consider a timer clock input of 32 kHz, with a PRE field equal to 0, the timer output period is:

Table 19-7. Value and Corresponding Interrupt Period

TLDR	Interrupt period
0000 0000h	37 h
FFFF 0000h	2 s
FFFF FFF0h	500 us
FFFF FFFEh	62.5 us

19.1.3.1.7 Dual Mode Timer Under Emulation

During emulation mode (when PINSUSPENDN signal is active), the timer can/cannot continue running according to the value of the EmuFree bit of the timer OCP configuration register (TIOCP_CFG).

If EmuFree is 1, timer execution is not stopped and, regardless of the value of PINSUSPENDN signal, and the interrupt assertion is still generated when overflow is reached.

If EmuFree is 0, counters (prescaler/timer) are frozen and an increment start occurs again as soon as PINSUSPENDN becomes inactive. The asynchronous input pin is internally synchronized on 2 TIMER clock rising edges.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.1.3.2 Accessing Registers

All registers are 32-bit wide, accessible via OCP interface with 16-bit or 32-bit OCP access (Read/Write). The 32-bit registers write update in 16 bits access must be LSB16 first and the second write access must be MSB16. For the write operation, the module allows skipping the MSB access if the user does not need to update the 16 MSB bits of the register, but only for the OCP registers (TIDR, TIOCP_CFG, IRQ_EOI, IRQSTATUS_RAW, IRQSTATUS, IRQENABLE_SET, IRQENABLE_CLR, IRQWAKEEN and TSICR). The write operation on any functional register (TCLR, TCRR, TLDR, TTGR and TMAR) must be complete (the MSB must be written even if the MSB data is not used).

19.1.3.2.1 Programming the Timer Registers

The TLDR, TCRR, TCLR, TIOCP_CFG, IRQ_EOI, IRQSTATUS, IRQENABLE_SET, IRQENABLE_CLR, IRQWAKEEN, TTGR, TSICR and TMAR registers write is done synchronously with OCP clock, by the host, using the OCP bus protocol.

19.1.3.2.2 Reading the Timer Registers

The counter register (TCRR) is a 32-bit “atomic datum” and 16-bit capture is done on the 16-bit LSB first to allow atomic LSB16 + MSB16 capture. Atomic capture is also performed for the TCAR1 and TCAR2 registers as they may change due to internal processes. DSP 16 bit accesses can be interleaved with MCU 32 bit accesses.

19.1.3.2.3 OCP Error Generation

The timer module responds with error indication in the following cases:

Error on write transactions

- Assert the PORSRESP = ERR signal in the same cycle as PORSCMDACCEPTED.
- Use the ERR code for PORSRESP during the response phase.

Error on read transactions

- Assert the PORSRESP = ERR signal in the same cycle as PORSCMDACCEPTED.
- Use the ERR code for PORSRESP during the response phase. PORSDATA in this case is not valid.

Table 19-8. OCP Error Reporting

Error Type	Response: SRESP = ERR
Unsupported PIOCPMCMD command	Yes
Address error: Read or write to a non-existing internal address	No
Read to write-only registers and write to read-only registers	No
Unaligned address (PIOCPMADDR ≠ 00) on read/write transaction	Yes
Unsupported PIOCPMBYTEEN on read/write transaction	Yes

NOTE: Byte enable “0000” is a supported byte enable.

19.1.3.3 Posted Mode Selection

A choice between the two synchronization modes will be made taking into account the frequency ratio and the stall periods that can be supported by the system, without impacting the global performance.

The posted mode selection applies only to functional registers that require synchronization on/from timer clock domain. For write operation the registers affected by this posted/non-posted selection are: TCLR, TLDR, TCRR, TTGR and TMAR. For read operation the register affected by this posted/non-posted selection are: TCRR, TCAR1 and TCAR2.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

The OCP clock domain synchronous registers TIDR, TIOCP_CFG, TISTAT, IRQ_EOI,IRQSTATUS,IRQSTATUS_RAW, IRQENABLE_SET, IRQENABLE_CLR, IRQWAKEEN, TWPS and TSICR are not affected by the posted/non-posted mode selection; the write/read operation is effective and acknowledged (command accepted) after one OCP clock cycle from the command assertion.

The configuration posted/non-posted is made by setting the PIFREQRATIO when the module is integrated. The PIFREQRATIO signal should be tied to '1' when the freq (timer) < freq (OCP)/4 frequency, and tied to 0 when it is the opposite frequency ratio. The PIFREQRATIO represent the reset value of the TSICR (POSTED bit). The configuration can be changed (overwritten) by software, writing the TSICR (POSTED bit) register.

The following cases are possible:

- Posted Mode can be used when the functional frequency range is: freq (timer) < freq (OCP)/4.
- Non-Posted Mode can be used regardless of the frequency range. Recommended frequency is: freq (timer) >= freq (OCP)/4.

NOTE: The Non-Posted Mode can be also used when freq (timer) < freq (OCP)/4, but it is recommended to use the Posted Mode. Using Non-Posted Mode will delay the command accept and the transaction latency will be as described in the below chapters. Posted mode offers an OCP interface latency improvement and can be used only if the frequencies respect the freq (timer) < freq (OCP)/4 formula.

19.1.3.4 Write Registers Access

19.1.3.4.1 Write Posted

This mode can be used only if the functional frequency range is freq (timer) < freq (OCP)/4.

This mode is used if TSICR (POSTED bit) is set to 1 in the timer control register.

This mode uses a posted-write scheme to update any internal register. The write transaction is immediately acknowledged on the OCP interface, although the effective write operation will occur later, due to a resynchronisation in the timer clock domain. This has the advantage of not stalling either the interconnect system, or the CPU that requested the write transaction. For each register, a status bit is provided, that is set if there is a pending write access to this register.

In this mode, it is mandatory that the CPU checks the status bit prior to any write access. In case a write is attempted to a register with a previous access pending, the previous access is discarded without notice (this can lead to unexpected results also).

There is one status bit per register, accessible in the Timer Write Posted Status Register. When the timer module operates in this mode, there is an automatic sampling of the current timer counter value, in an OCP-synchronized capture register. Consequently, any read access to the timer counter register does not add any re-synchronization latency; the current value is always available.

A register read following a write posted register (on the same register) is not insured to read the previous write value if the write posted process is not completed. Software synchronization should be used to avoid non-coherent read.

The drawback of this automatic update mechanism is that it assumes a given relationship between the OCP interface frequency and the timer functional frequency.

This posted period is defined as the interval between the posted write access request and the reset of the posted bit in TWPS register, and can be quantified:

T (reset posted max.) = 3 OCP clock + 2.5 TIMER clock

The time when the write accomplishes is:

T (write accomplish) = 1 OCP clock + 2.5 TIMER clock

Copyright © 2011, Texas Instruments Incorporated

19.1.3.4.2 Write Non-Posted

This mode is functional regardless of the ratio between the OCP interface frequency and the functional clock frequency. Recommended functional frequency range is freq (timer) >= freq (OCP)/4.

This mode is used if TSICR (POSTED bit) is cleared to 0 in the timer control register.

This mode uses a non posted-write scheme to update any internal register. That means the write transaction will not be acknowledged on the OCP interface, until the effective write operation occurs, after the resynchronisation in the timer clock domain. The drawback is that both the interconnect system and the CPU are stalled during this period.

- The latency of the interrupt serving is increased, as the interconnect system and CPU are stalled.
- An interconnect logic, including time-out logic to detect erroneous transactions, can generate an unwanted system abort event.

The stall period is defined as the interval between the non-posted write access request and the rise of the command accept signal and can be quantified:

T (stall max.) = 3 OCP clock + 2.5 TIMER clock

The time when the write accomplishes is:

T (write accomplish) = 1 OCP clock + 2.5 TIMER clock

A register read following a write to the same register is always coherent.

19.1.3.5 Read Registers Access

19.1.3.5.1 Read Posted

This mode can be used only if the functional frequency range is freq (timer) < freq (OCP)/4.

This mode is used if TSICR (POSTED bit) is set to 1 in the timer control register.

This mode uses a posted-read scheme, for reading any internal register. The read transaction is immediately acknowledged on the OCP interface, and the value to be read has been previously resynchronised. This has the advantage of not stalling either the interconnect system, or the CPU that requested the read transaction.

19.1.3.5.2 Read Non-Posted

This mode is functional whatever the ratio between the OCP interface frequency and the functional clock frequency. Recommended functional frequency range is freq (timer) >= freq (OCP)/4.

This mode is used if TSICR (POSTED bit) is cleared to 0 in the timer control register.

This mode uses a non posted-read scheme, for reading any internal register. The read transaction will not be acknowledged on the OCP interface, until the effective read operation occurs, after the resynchronisation in the timer clock domain. The drawback is that both the interconnect system and the CPU are stalled during this period.

- The latency of the interrupt serving is increased, as the interconnect system and the CPU are stalled.
- An interconnect system including time-out logic to detect erroneous transactions can generate an unwanted system abort event.

This mode only applies only to three registers: TCRR, TCAR1 and TCAR2, which need resynchronisation from functional to OCP clock domains.

The stall period is defined as the interval between the non-posted read access request and the rise of the command accept signal and can be quantified:

T (stall max.) = 3 OCP clock + 2.5 TIMER clock

The time when the value is sampled is:

T (read sample) = 1 OCP clock + 2.5 TIMER clock

19.1.4 Use Cases

Copyright © 2011, Texas Instruments Incorporated

19.1.5 Timer Registers

The timer registers are listed in Table 19-9. All registers are:

- 32-bit register accessible in 16-bit mode
- Little-endian addressing

Table 19-9. Timer Registers

Address Offset	Acronym	Register Name
00h	TIDR	Identification Register
10h	TIOCP_CFG	Timer OCP Configuration Register
20h	IRQ_EOI	Timer IRQ End-Of-Interrupt Register
24h	IRQSTATUS_RAW	Timer IRQSTATUS Raw Register
28h	IRQSTATUS	Timer IRQSTATUS Register
2Ch	IRQSTATUS_SET	Timer IRQENABLE Set Register
30h	IRQSTATUS_CLR	Timer IRQENABLE Clear Register
34h	IRQWAKEEN	Timer IRQ Wakeup Enable Register
38h	TCLR	Timer Control Register
3Ch	TCRR	Timer Counter Register
40h	TLDR	Timer Load Register
44h	TTGR	Timer Trigger Register
48h	TWPS	Timer Write Posted Status Register
4Ch	TMAR	Timer Match Register
50h	TCAR1	Timer Capture Register
54h	TSICR	Timer Synchronous Interface Control Register
58h	TCAR2	Timer Capture Register

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.1.5.1 Identification Register (TIDR)

This read only register contains the revision number of the module. A write to this register has no effect. This register is used by software to track features, bugs, and compatibility.

Figure 19-7. Identification Register (TIDR) Register

31–30	29–28	27–16
SCHEME	Reserved	FUNC
R/W-1	R-0	R/W-0

15–1	10–9–8	7–6	5–0
R_RTL	X_MAJOR	CUSTOM	Y_MINOR
R/W-0	R/W-1	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-10. Identification Register (TIDR) Field Descriptions

Bit	Field	Value	Description
31-30	SCHEME	1	Used to distinguish between old scheme and current.
29-28	Reserved	0	Reads return 0
27-16	FUNC	0	Function indicates a software compatible module family. If there is no level of software compatibility a new Func number (and hence REVISION) should be assigned.
15-11	R_RTL	0	RTL Version (R), maintained by IP design owner. RTL follows a numbering such as X.Y.R.Z which are explained in this table. R changes ONLY when: (1) PDS uploads occur which may have been due to spec changes (2) Bug fixes occur (3) Resets to '0' when X or Y changes. Design team has an internal 'Z' (customer invisible) number which increments on every drop that happens due to DV and RTL updates. Z resets to 0 when R increments.
10-8	X_MAJOR	1	Major Revision (X), maintained by IP specification owner. X changes ONLY when: (1) There is a major feature addition. An example would be adding Master Mode to Utopia Level2. The Func field (or Class/Type in old PID format) will remain the same. X does NOT change due to: (1) Bug fixes (2) Change in feature parameters.
7-6	CUSTOM	0	Indicates a special version for a particular device. Consequence of use may avoid use of standard Chip Support Library (CSL) / Drivers.
5-0	Y_MINOR	0	Minor Revision (Y), maintained by IP specification owner. Y changes ONLY when: (1) Features are scaled (up or down). Flexibility exists in that this feature scalability may either be represented in the Y change or a specific register in the IP that indicates which features are exactly available. (2) When feature creeps from Is-Not list to Is list. But this may not be the case once it sees silicon; in which case X will change. Y does NOT change due to: (1) Bug fixes (2) Typos or clarifications (3) major functional/feature change/addition/deletion. Instead these changes may be reflected via R, S, X as applicable. Spec owner maintains a customer-invisible number 'S' which changes due to: (1) Typos/clarifications (2) Bug documentation. Note that this bug is not due to a spec change but due to implementation. Nevertheless, the spec tracks the IP bugs. An RTL release (say for silicon PG1.1) that occurs due to bug fix should document the corresponding spec number (X.Y.S) in its release notes.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.1.5.2 Timer OCP Configuration Register (TIOCP_CFG)

This register allows controlling various parameters of the OCP interface.

Figure 19-8. Timer OCP Configuration Register (TIOCP_CFG)

31 – 4	3 – 2	1	0
Reserved	IDLEMODE	EMUFREE	SOFTRESET
R-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-11. Timer OCP Configuration Register (TIOCP_CFG) Field Descriptions

Bit	Field	Value	Description
31-4	Reserved	0	Reserved
3-2	IDLEMODE		Power management, req/ack control
		0	Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, i.e. regardless of the IP module's internal requirements. Backup mode, for debug only.
		1h	No-idle mode: local target never enters idle state. Backup mode, for debug only.
		2h	Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module shall not generate (IRQ- or DMA-request-related) wakeup events.
		3h	Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module may generate (IRQ- or DMA-request-related) wakeup events when in idle state. Mode is only relevant if the appropriate IP module "swakeup" output(s) is (are) implemented.
1	EMUFREE		Emulation mode
		0	The timer is frozen in emulation mode (PINSUSPENDN signal active).
		1	The timer runs free, regardless of PINSUSPENDN value.
0	SOFTRESET		Software reset
		R0	Reset done, no pending action
		W0	No action
		R1	Initiate software reset
		W1	Reset ongoing

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.1.5.3 Timer IRQ EOI Register (IRQ_EOI)

Software End-Of-Interrupt: Allows the generation of further pulses on the interrupt line, if an new interrupt event is pending, when using the pulsed output.

Unused when using the level interrupt line (depending on module integration).

Figure 19-9. Timer IRQ EOI Register (IRQ_EOI)

31 – 1	4 – 0
Reserved	LINE_NUMBER
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-12. Timer IRQ EOI Register (IRQ_EOI) Field Descriptions

Bit	Field	Value	Description
31-1	Reserved	0	Reserved
0	LINE_NUMBER		Write the number of the interrupt line to apply a SW EOI to it.
			Note that there is only a single line (that is, number 0)
		R0	Always returns 0
		W0	SW EOI on interrupt line
		W1	No action

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.1.5.4 Timer IRQSTATUS Raw Register (IRQSTATUS_RAW)

Component interrupt request status.

Check the corresponding secondary status register. Raw status is set even if event is not enabled. Write 1 to set the (raw) status, mostly for debug.

Figure 19-10. Timer IRQSTATUS Raw Register (IRQSTATUS_RAW)

31				16
Reserved				
R-0				

15	3	2	1	0
Reserved		TCAR_IT_FLAG	OVF_IT_FLAG	MAT_IT_FLAG
R-0		R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-13. Timer IRQSTATUS Raw Register (IRQSTATUS_RAW) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Reserved
2	TCAR_IT_FLAG		IRQ status for Capture
		R0	No event pending
		W0	No action
		R1	IRQ event pending
		W1	Trigger IRQ event by software
1	OVF_IT_FLAG		IRQ status for Overflow
		R0	No event pending
		W0	No action
		R1	IRQ event pending
		W1	Trigger IRQ event by software
0	MAT_IT_FLAG		IRQ status for Match
		R0	No event pending
		W0	No action
		R1	IRQ event pending
		W1	Trigger IRQ event by software

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.1.5.5 Timer IRQSTATUS Register (IRQSTATUS)

Component interrupt request status.

Check the corresponding secondary status register. Enabled status is not set unless event is enabled. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled).

Figure 19-11. Timer IRQSTATUS Register (IRQSTATUS)

31				16
Reserved				
R-0				

15	3	2	1	0
Reserved		TCAR_IT_FLAG	OVF_IT_FLAG	MAT_IT_FLAG
R-0		R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-14. Timer IRQSTATUS Register (IRQSTATUS) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Reserved
2	TCAR_IT_FLAG		IRQ status for Capture
		R0	No event pending
		W0	No action
		R1	IRQ event pending
		W1	Clear pending event, if any
1	OVF_IT_FLAG		IRQ status for Overflow
		R0	No event pending
		W0	No action
		R1	IRQ event pending
		W1	Clear pending event, if any
0	MAT_IT_FLAG		IRQ status for Match
		R0	No event pending
		W0	No action
		R1	IRQ event pending
		W1	Clear pending event, if any

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.1.5.6 Timer IRQENABLE Set Register (IRQENABLE_SET)

Component interrupt request enable.

Write 1 to set (enable interrupt).

Readout equal to corresponding _CLR register.

Figure 19-12. 3Timer IRQENABLE Set Register (IRQENABLE_SET)

31				16
Reserved				
R-0				

15	3	2	1	0
Reserved		TCAR_EN_FLAG	OVF_EN_FLAG	MAT_EN_FLAG
R-0		R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-15. Timer IRQENABLE Set Register (IRQENABLE_SET) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Reserved
2	TCAR_EN_FLAG		IRQ enable for Compare
		R0	IRQ event is disabled
		W0	No action
		R1	IRQ event is enabled
		W1	Clear IRQ enable
1	OVF_EN_FLAG		IRQ enable for Overflow
		R0	IRQ event is disabled
		W0	No action
		R1	IRQ event is enabled
		W1	Clear IRQ enable
0	MAT_EN_FLAG		IRQ enable for Match
		R0	IRQ event is disabled
		W0	No action
		R1	IRQ event is enabled
		W1	Clear IRQ enable

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.1.5.7 Timer IRQENABLE Clear Register (IRQENABLE_CLR)

Component interrupt request enable.

Write 1 to clear (disable interrupt).

Readout equal to corresponding _SET register.

Figure 19-13. Timer IRQENABLE Clear Register (IRQENABLE_CLR)

31				16
Reserved				
R-0				

15	3	2	1	0
Reserved		TCAR_EN_FLAG	OVF_EN_FLAG	MAT_EN_FLAG
R-0		R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-16. Timer IRQENABLE Clear Register (IRQENABLE_CLR) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Reserved
2	TCAR_EN_FLAG		IRQ enable for Compare
		R0	IRQ event is disabled
		W0	No action
		R1	IRQ event is enabled
		W1	Clear IRQ enable
1	OVF_EN_FLAG		IRQ enable for Overflow
		R0	IRQ event is disabled
		W0	No action
		R1	IRQ event is enabled
		W1	Clear IRQ enable
0	MAT_EN_FLAG		IRQ enable for Match
		R0	IRQ event is disabled
		W0	No action
		R1	IRQ event is enabled
		W1	Clear IRQ enable

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.1.5.8 Timer IRQ Wakeup Enable Register (IRQWAKEEN)

Wakeup-enabled events taking place when module is idle will generate an asynchronous wakeup.

Figure 19-14. Timer IRQ Wakeup Enable Register (IRQWAKEEN)

31–16	15–3	2	1	0
Reserved	Reserved	TCAR_WUP_FLAG	OVF_WUP_FLAG	MAT_WUP_FLAG
R-0	R-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-17. Timer IRQ Wakeup Enable Register (IRQWAKEEN) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Reserved
2	TCAR_WUP_ENA		Wakeup generation for Compare
		0	Wakeup disabled
		1	Wakeup enabled
1	OVF_WUP_ENA		Wakeup generation for Overflow
		0	Wakeup disabled
		1	Wakeup enabled
0	MAT_WUP_ENA		Wakeup generation for Match
		0	Wakeup disabled
		1	Wakeup enabled

19.1.5.9 Timer Control Register (TCLR)

NOTE: When the TCM field passed from (00) to any other combination then the TCAR_IT_FLAG and the edge detection logic are cleared.

The ST bit of TCLR register may be updated from the OCP interface or reset due to one-shot overflow. The OCP interface update has the priority.

Figure 19-15. Timer Control Register (TCLR)

31–16	15	14	13	12	11–10	9–8
Reserved	Reserved	GPO_CFG	CAPT_MODE	PT	TRG	TCM
R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

7	6	5	4–2	1	0
SCPWM	CE	PRE	PTV	AR	ST
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 19-18. Timer Control Register (TCLR) Field Descriptions

Bit	Field	Value	Description
31-15	Reserved	0	Reserved
14	GPO_CFG		General purpose output—this register drives directly the PORGPOCFG output pin
		0	PORGPOCFG drives 0
		1	PORGPOCFG drives 1
13	CAPT_MODE	0	Single capture
		1	Capture on second event
12	PT		Pulse or toggle mode on PORTIMERPWM output pin
		0	Pulse
		1	Toggle
11-10	TRG		Trigger output mode on PORTIMERPWM output pin
		0	No trigger
		1h	Trigger on overflow
		2h	Trigger on overflow and match
		3h	Reserved
9-8	TCM		Transition Capture Mode on PIEVENTCAPT input pin
		0	No capture
		1h	Capture on low to high transition
		2h	Capture on high to low transition
		3h	Capture on both edge transition
7	SCPWM		This bit should be set or clear while the timer is stopped or the trigger is off
		0	Clear the PORTIMERPWM output pin and select positive pulse for pulse mode
		1	Set the PORTIMERPWM output pin and select negative pulse for pulse mode
6	CE	0	Compare mode is disabled
		1	Compare mode is enabled
5	PRE		Prescaler enable
		0	The TIMER clock input pin clocks the counter
		1	The divided input pin clocks the counter
4-2	PTV	0	Pre-scale clock Timer value
1	AR	0	One shot timer
		1	Auto-reload timer
0	ST		In the case of one-shot mode selected (AR = 0), this bit is automatically reset by internal logic when the counter is overflowed.
		0	Stop timer: Only the counter is frozen
		1	Start timer

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.1.5.10 Timer Counter Register (TCRR)

The TCRR register is a 32-bit register, 16-bit addressable. The MCU can perform a 32-bit access or two 16-bit accesses to access the register while the DSP performs 2 consecutive 16-bit transactions. Note that since the OCP clock is completely asynchronous with the timer clock, some synchronization is done in order to make sure that the TCRR value is not read while it is being incremented.

In 16-bit mode the following sequence must be followed to read the TCRR register properly:

- Perform an OCP Read Transaction to Read the lower 16-bit of the TCRR register (offset = 28h). When the TCRR is read and synchronized, the lower 16-bit LSBs are driven onto the output OCP data bus and the upper 16-bit MSBs of the TCRR register are stored in a temporary register.
- Perform an OCP Read Transaction to read the upper 16-bit of the TCRR register (offset = 2Ah). During this Read, the value of the upper 16-bit MSBs that has been temporary register is forwarded onto the output OCP data bus.

So, to read the value of TCRR correctly, the first OCP read access has to be to the lower 16-bit (offset = 28h), followed by OCP read access to the upper 16-bit (offset = 2Ah).

As the TCRR is updated using more sources (shadow_in_tcrr, incremented value of tcrr, TLDR and “0”), a priority order will be defined:

- The first priority is the OCP update
- The second is the reload way (triggered through TTGR reg. or following an auto-reload overflow)
- The third is the one-shot overflow reset to 0
- The last is the incremented value

Figure 19-16. Timer Counter Register (TCRR)

31	0
TIMER_COUNTER	
R-0	

LEGEND: R = Read only; -*n* = value after reset

Table 19-19. Timer Counter Register (TCRR) Field Descriptions

Bit	Field	Value	Description
31-0	TIMER_COUNTER	0	Value of TIMER counter

19.1.5.11 Timer Load Register (TLDR)

NOTE: LOAD_VALUE must be different than the timer overflow value (FFFF FFFFh).

Figure 19-17. Timer Load Register (TLDR)

31	0
LOAD_VALUE	
R-0	

LEGEND: R = Read only; -*n* = value after reset

Table 19-20. Timer Load Register (TLDR) Field Descriptions

Bit	Field	Value	Description
31-0	LOAD_VALUE	0h	Timer counter value loaded on overflow in auto-reload mode or on TTGR write access

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.1.5.12 Timer Trigger Register (TTGR)

The read value of this register is always FFFF FFFFh.

Figure 19-18. Timer Trigger Register (TTGR)

31	0
TTGR_VALUE	
R-FFFF FFFFh	

LEGEND: R = Read only; -*n* = value after reset

Table 19-21. Timer Trigger Register (TTGR) Field Descriptions

Bit	Field	Value	Description
31-0	TTGR_VALUE	FFFF FFFFh	Writing in the TTGR register, TCRR will be loaded from TLDR and prescaler counter will be cleared Reload will be done regardless of the AR field value of TCLR register.

19.1.5.13 Timer Write Posted Status Register (TWPS)

In posted mode the software must read the pending write status bits (Timer Write Posted Status register bits [4:0]) to insure that following write access will not be discarded due to on going write synchronization process. These bits are automatically cleared by internal logic when the write to the corresponding register is acknowledged.

Figure 19-19. Timer Write Posted Status Register (TWPS)

31	16
Reserved	
R-0	

15	5	4	3	2	1	0
Reserved		W_PEND_TMAR	W_PEND_TTGR	W_PEND_TLDR	W_PEND_TCRR	W_PEND_TCLR
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-22. Timer Write Posted Status Register (TWPS) Field Descriptions

Bit	Field	Value	Description
31-5	Reserved	0	Reserved
4	W_PEND_TMAR	0	When equal to 1, a write is pending to the TMAR register
3	W_PEND_TTGR	0	When equal to 1, a write is pending to the TTGR register
2	W_PEND_TLDR	0	When equal to 1, a write is pending to the TLDR register
1	W_PEND_TCRR	0	When equal to 1, a write is pending to the TCRR register
0	W_PEND_TCLR	0	When equal to 1, a write is pending to the TCLR register

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.1.5.14 Timer Match Register (TMAR)

The compare logic consists of a 32-bit wide, read/write data TMAR register and logic to compare counter's current value with the value stored in the TMAR register.

Figure 19-20. Timer Match Register (TMAR)

31	0
COMPARE_VALUE	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 19-23. Timer Match Register (TMAR) Field Descriptions

Bit	Field	Value	Description
31-0	COMPARE_VALUE	0	Value to be compared to the timer counter

19.1.5.15 Timer Capture Register (TCAR1)

When the appropriate (rising, falling or both) transition is detected in the edge detection logic the current counter value is stored to the TCAR1 register. Note that since the OCP clock is completely asynchronous with the timer clock, some synchronization is done in order to make sure that the TCAR1 value is not read while it is being updated due to some capture event.

In 16-bit mode the following sequence must be followed to read the TCAR1 register properly:

- Perform an OCP Read Transaction to Read the lower 16-bits of the TCAR1 register.
- Perform an OCP Read Transaction to read the upper 16-bits of the TCAR1 register.

Figure 19-21. Timer Capture Register (TCAR1)

31	0
CAPTURED_VALUE	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 19-24. Timer Capture Register (TCAR1) Field Descriptions

Bit	Field	Value	Description
31-0	CAPTURED_VALUE	0	Timer counter value captured on an external event trigger

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.1.5.16 Timer Synchronous Interface Control Register (TSICR)

Access to this register is not stalled even if the timer is in non-posted mode configuration. To abort any wrong behavior, software can permanently reset the functional part of the module. Also in case of a wrong hardware PIFREQRATIO tied the POSTED field can be reprogrammed on the fly, so deadlock situation cannot happen.

NOTE: Reset value of POSTED depends on hardware integration module at design time. Software must read POSTED field to get the hardware module configuration.

Figure 19-22. Timer Synchronous Interface Control Register (TSICR)

31	3	2	1	0
Reserved		POSTED	SFT	Reserved
R-0		R/W-0	R/W-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-25. Timer Synchronous Interface Control Register (TSICR) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Reserved
2	POSTED		PIFREQRATIO
		0	Posted mode inactive: will delay the command accept output signal.
		1	Posted mode active (clocks ratio needs to fit freq (timer) < freq (OCP)/4 frequency requirement)
1	SFT		This bit resets all the function parts of the module. During reads it always returns 0.
		0	Software reset is enabled
		1	Software reset is disabled
0	Reserved	0	Reserved

19.1.5.17 Timer Capture Register (TCAR2)

When the appropriate (rising, falling or both) transition is detected in the edge detection logic and the capture on second event is activated from the control register (TCLR), the current counter value is stored to the TCAR2 register. Note that since the OCP clock is completely asynchronous with the timer clock, some synchronization is done in order to make sure that the TCAR2 value is not read while it is being updated due to some capture event.

In 16-bit mode the following sequence must be followed to read the TCAR2 register properly:

- Perform an OCP Read Transaction to Read the lower 16-bits of the TCAR2 register.
- Perform an OCP Read Transaction to read the upper 16-bits of the TCAR2 register.

Figure 19-23. Timer Capture Register (TCAR2)

31	0
CAPTURED_VALUE	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 19-26. Timer Capture Register (TCAR2) Field Descriptions

Bit	Field	Value	Description
31-0	CAPTURED_VALUE	0	Timer counter value captured on an external event trigger

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.2 DMTimer 1ms

19.2.1 Introduction

This peripheral is a 32-bit timer offering:

- Counter timer with compare and capture modes
- Auto-reload mode
- Start-stop mode
- Generate 1 ms tick with 32768-Hz functional clock
- Programmable divider clock source
- 16–32 bit addressing
- On-the-fly read/write registers
- Interrupts generated on overflow, compare and capture
- Interrupt enable
- Wake-up enable
- Write posted mode
- Dedicated input trigger for capture mode and dedicated output trigger/PWM signal
- Dedicated output signal for general purpose use PORGPOCFG
- OCP interface compatible

The timer module contains a free running upward counter with auto reload capability on overflow. The timer counter can be read and written on the fly (while counting). The timer module includes compare logic to allow interrupt event on programmable counter matching value.

A dedicated output signal can be pulsed or toggled on overflow and match event. This output offers timing stamp trigger signal or PWM (pulse width modulation) signal sources. A dedicated output signal can be used for general purpose PORGPOCFG (directly driven by the bit 14 of the TCLR register). A dedicated input signal is used to trigger automatic timer counter capture and interrupt event, on programmable input signal transition type. A programmable clock divider (prescaler) allows reduction of the timer input clock frequency. All internal timer interrupt sources are merged in one module interrupt line and one wake-up line. Each internal interrupt sources can be independently enabled/disabled with a dedicated bit of TIER register for the interrupt features and a dedicated bit of TWER for the wake-up.

This module is controllable through the OCP peripheral bus.

As 2 clocks domains are managed inside this module, resynchronization is done by special logic between OCP clock domain and Timer clock domain. At reset, synchronization logic allows utilization of all ratios between OCP clock and Timer clock. Drawback of this mode is that full-resynchronization path is used with access latency performance impact in terms of OCP clock cycles. In order to improve module access latency, and under restricted conditions on clocks ratios (cf. 7.1 Write posted), write-posted mode can be used by setting POSTED bit of System Control register (TSICR). Under this mode, write posted mode is enabled, meaning that OCP write command is granted before write process complete in the timer clock domain. This mode allows software (SW) to do concurrent writes on Dual Mode timer registers and to observe write process completion (synchronization) at SW level by reading independent write posted status bits in the Write Posted Status Register (TWPS).

Figure 19-24. Block Diagram

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.2.2 Integration

19.2.2.1 Timer Connectivity Attributes

Table 19-27. Timer1 Connectivity Attributes

Attributes	Type
Power Domain	Wakeup Domain
Clock Domain	PD_WKUP_L4_WKUP_GCLK (OCP) PD_WKUP_TIMER1_GCLK (Func)
Reset Signals	WKUP_DOM_RST_N
Idle/Wakeup Signals	Smart Idle / Slave Wakeup
Interrupt Requests	1 to Sabertooth, WakeM3
DMA Requests	None
Physical Address	L4 Wakeup slave port

19.2.2.2 Timer Clock and Reset Manangement

The DMTimer1 1ms timer functional clock can be selected from one of five sources using the CLKSEL_TIMER1MS_CLK register in the PRCM:

- The 24 MHz (typ) system clock (CLK_M_OSC)
- The PER PLL generated 32.768 KHz clock (CLK_32KHZ)
- The TCLKIN external timer input clock
- The on-chip ~32.768 KHz oscillator (CLK_RC32K)
- The external 32.768 KHz oscillator/clock (CLK_32K_RTC)

19.2.2.3 Timer Clock Signals

Table 19-28. Timer Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
Timer1 (1ms) Clock Signals			
PICLKOCP Interface clock	100 MHz	CORE_CLKOUTM4 / 2	pd_wkup_l4_wkup_gclk from PRCM
PICLKTIMER Functional clock	25 MHz[(1)]	CLK_M_OSC CLK_32KHZ (PER_CLKOUTM2 / 5859.375) TCLKIN CLK_RC32K CLK_32K_RTC	pd_wkup_timer1_gclk from PRCM

(1) Max frequency depends on final SOC timing closure when TCLKIN is selected as the clock source. But PICLKTIMER must be less than PICLKOCP/4.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.2.3 Functional Description

The general-purpose timer is an upward counter. It supports 3 functional modes:

- Timer mode
- Capture mode
- Compare mode

By default, after core reset, the capture and compare modes are disabled.

19.2.3.1 Timer Mode Functionality

The timer is an upward counter that can be started and stopped at any time through the Timer Control Register (TCLR ST bit). The Timer Counter Register (TCRR) can be loaded when stopped or on the fly (while counting). TCRR can be loaded directly by a TCRR Write access with the new timer value. TCRR can also be loaded with the value held in the Timer Load Register TLDR by a trigger register (TTGR) Write access. The TCRR loading is done regardless the TTGR written value. The timer counter register TCRR value can be read when stopped or captured on the fly by a TCRR Read access. The timer is stopped and the counter value set to “0” when the module’s reset is asserted. The timer is maintained in stop after reset is released. When the timer is stopped TCRR is frozen and it can be restarted from the frozen value unless TCRR has been reloaded with a new value.

In the one shot mode (TCLR AR bit =”0”), the counter is stopped after counting overflow (counter value remains at zero).

When the auto-reload mode is enabled (TCLR AR bit =”1”), the TCRR is reloaded with the Timer Load Register (TLDR) value after a counting overflow.

It is not recommended to put the overflow value (0xFFFFFFFF) in TLDR because it can lead to undesired results.

An interrupt can be issued on overflow if the overflow interrupt enable bit is set in the timer Interrupt Enable Register (TIER OVF_IT_ENA bit =”1”). A dedicated output pin (PORTIMERPWM) is programmed through TCLR (TRG and PT bits) to generate one positive pulse (prescaler duration) or to invert the current value (toggle mode) when an overflow occurs.

Figure 19-25. TCRR Timing Value

19.2.3.1.1 1 ms Tick Generation

To minimize the error between a true 1ms tick and the tick generated by the 32768 Hz timer the sequencing of the sub-1ms periods and the over-1ms periods must be shuffled.

An additional block (1ms block) is used to correct this error.

In this implementation the increment sequencing is automatically managed by the timer to minimize the error. The value of the Timer Positive Increment register (TPIR) and Timer Negative Increment register (TNIR) only need to be defined by the user. Auto adaptation mechanism is used to simplify the programming model.

Figure 19-26. 1 ms Module Block Diagram

The TPIR, TNIR, TCVR and adders Add1..3 are used to define whether next value loaded in the TCRR will be value of the TLDR (sub-period value) or the value of TLDR – 1 (over-period value).

The following table shows the value loaded in TCRR according to the sign of the result of Add1, Add2 and Add3. MSB = ‘0’ means a positive value, MSB = ‘1’ means a negative value.

Table 19-29. Value Loaded in TCRR to Generate 1 ms Tick

Add1 MSB	Add2 MSB	Add3 MSB	TCRR
0	0	0	TLDR
0	0	1	TLDR

Copyright © 2011, Texas Instruments Incorporated

Table 19-29. Value Loaded in TCRR to Generate 1 ms Tick (continued)

Add1 MSB	Add2 MSB	Add3 MSB	TCRR
0	1	0	TLDR
0	1	1	TLDR - 1
1	0	0	N.A.
1	0	1	N.A.
1	1	0	TLDR - 1
1	1	1	TLDR - 1

The values of TPIR and TNIR registers are calculated with formula:

Positive Increment Value = ((INTEGER[Fclk * Ttick] + 1) * 1e6) - (Fclk * Ttick * 1e6)

Negative Increment Value = (INTEGER[Fclk * Ttick] * 1e6) - (Fclk * Ttick * 1e6)

where:

Fclk – clock frequency (KHz)

Ttick – tick period (ms)

The Timer Overflow Counter Register (TOCR) and the Timer Overflow Wrapping Register (TOWR) are used for interrupt filtering. When timer overflows it increments the 24 bit TOCR register. When 24 bit TOCR register values matches the value in the 24 bit TOWR register and timer overflow is asserted, the TOCR is reset and an interrupt is generated to TISR.

With the Conversion block, in reset state (Positive Increment register, Negative Increment register and Counter Value register are all zeroed) the programming model and the behavior of the DMtimer_dmc1ms remain unchanged.

For 1 ms tick with a 32768-Hz clock:

TPIR = 232000

TNIR = -768000

TLDR = 0xFFFFFFE0

NOTE: Any value of the tick period can be generated with appropriate value of the TPIR, TNIR and TLDR registers.

By default the TPIR, TNIR, TCVR, TOCR, TOWR registers and the associated logic are in reset mode (all 0s) and have no action on the programming model of the DMtimer_dmc1ms.

19.2.3.2 Capture Mode Functionality

The timer value in TCRR can be captured and saved in TCAR1 or TCAR2 function of the mode selected in TCLR through the field CAPT_MODE when a transition is detected on the module input pin (PIEVENTCAPT). The edge detection circuitry monitors transitions on the input pin (PIEVENTCAPT).

Rising transition, falling transition or both can be selected in TCLR (TCM bit) to trig the timer counter capture. The module sets the TISR (TCAR_IT_FLAG bit) when an active transition is detected and at the same time the counter value TCRR is stored in one of the timer capture registers TCAR1 or TCAR2 as follows:

- If TCLR's CAPT_MODE field is "0" then, on the first enabled capture event, the value of the counter register is saved in TCAR1 register and all the next events are ignored (no update on TCAR1 and no interrupt triggering) until the detection logic is reset or the interrupt status register is cleared on TCAR's position writing a "1" in it. .
- If TCLR's CAPT_MODE field is "1" then, on the first enabled captured event, the counter value is saved in TCAR1 register and, on the second enabled capture event, the value of the counter register is saved in TCAR2 register. If capture interrupt is enabled, the interrupt will be asserted on the second event capture. All the other events are ignored (no update on TCAR1/2 and no interrupt triggering) until the detection logic is reset or the interrupt status register is cleared on TCAR's position writing a "1" in it. This mechanism is useful for period calculation of a clock if that clock is connected to the

Copyright © 2011, Texas Instruments Incorporated

PIEVENTCAPT input pin.

The edge detection logic is reset (a new capture is enabled) when the active capture interrupt is served - TCAR_IT_FLAG bit of TISR (previously '1') is cleared through a "1" written in it or when the edge detection mode bits TCLR (TCM bit) passed from the No Capture Mode detection to any other modes. The timer functional clock (input to prescaler) is used to sample the input pin (PIEVENTCAPT). Input negative or positive pulse can be detected when pulse time is above functional clock period. An interrupt can be issued on transition detection if the capture interrupt enable bit is set in the Timer Interrupt Enable Register TIER (TCAR_IT_ENA bit).

See the following examples:

In the next wave, the TCM value is "01" and CAPT_MODE is "0"- only rising edge of the PIEVENTCAPT will trigger a capture in TCAR and only TCAR1 will update.

Figure 19-27. Capture Wave Example for CAPT_MODE 0

In the following example, the TCM value is "01" and CAPT_MODE is "1"- only rising edge of the PIEVENTCAPT will trigger a capture in TCAR1 on first enabled event and TCAR2 will update on the second enabled event.

Figure 19-28. Capture Wave Example for CAPT_MODE 1

19.2.3.3 Compare Mode Functionality

When Compare Enable TCLR (CE bit) is set to "1", the timer value (TCRR) is permanently compared to the value held in timer match register (TMAR). TMAR value can be loaded at any time (timer counting or stop). When the TCRR and the TMAR values match, an interrupt can be issued if the TIER (MAT_IT_ENA bit) is set. The right programming way is to write a compare value in TMAR register before setting TCLR (CE bit) to avoid any unwanted interrupt due to a reset value matching effect.

The dedicated output pin (PORTIMERPWM) can be programmed through TCLR (TRG and PT bits) to generate one positive pulse (TIMER clock duration) or to invert the current value (toggle mode) when an overflow and a match occur.

19.2.3.4 Prescaler Functionality

A prescaler counter can be used to divide the timer counter input clock frequency. The prescaler is enabled when TCLR bit 5 is set (PRE). The 2n division ratio value (PTV) can be configured in the TCLR register.

The prescaler counter is reset when the timer counter is stopped or reloaded on the fly.

Table 19-30. Prescaler/Timer Reload Values Versus Contexts

Contexts	Prescaler Counter	Timer Counter
Overflow (when Auto-reload on)	reset	TLDR
TCRR Write	reset	TCRR
TTGR Write	reset	TLDR
Stop	reset	Frozen

19.2.3.5 Pulse-Width Modulation

The timer can be configured to provide a programmable pulse-width modulation (PORTIMERPWM) output. The PORTIMERPWM output pin can be configured to toggle on specified event. TCLR (TRG bits) determines on which register value the PORTIMERPWM pin toggles. Either overflow or match can be used to toggle the PORTIMERPWM pin, when a compare condition occurs.

In case of overflow and match mode, the match event will be ignored from the moment the mode was set-up until the first overflow event occurs

Copyright © 2011, Texas Instruments Incorporated

The TCLR (SCPWM bit) can be programmed to set or clear the PORTIMERPWM output signal while the counter is stopped or the triggering is off only. This allows fixing a deterministic state of the output pin when modulation is stopped. The modulation is synchronously stopped when TRG bit is cleared and overflow occurred.

In the following timing diagram, the internal overflow pulse is set each time (0xFFFF FFFFF – TLDR +1) value is reached, and the internal match pulse is set when the counter reaches TMAR register value. According to TCLR (TRG and PT bits) programming value, the timer provides pulse or PWM on the output pin (PORTIMERPWM).

The TLDR and TMAR registers must keep values smaller than the overflow value (0xFFFFFFFF) with at least 2 units. In case the PWM trigger events are both overflow and match, the difference between the values kept in TMAR register and the value in TLDR must be at least 2 units. When match event is used the compare mode TCLR (CE) must be set.

On the following wave TCLR (SCPWM bit) is set to ‘0’.

Figure 19-29. Timing Diagram of Pulse-Width Modulation, SCPWM Bit = 0

On the next wave TCLR (SCPWM bit) is set to ‘1’.

Figure 19-30. Timing Diagram of Pulse-Width Modulation, SCPWM Bit = 1

19.2.3.6 Timer Interrupt Control

The timer can issue an overflow interrupt, a timer match interrupt and a timer capture interrupt. Each internal interrupt sources can be independently enabled/disabled in the Interrupt Enable Register TIER. When the interrupt event has been issued the associated interrupt status bit is set in the Timer Status Register (TISR). The pending interrupt event is reset when the set status bit is overwritten by a “1” value. Reading the Interrupt Status Register and writing the value back allows fast acknowledge interrupt process.

19.2.3.7 Sleep Mode Request and Acknowledge

Upon a Sleep mode request issued by the host processor (the Idle Request PIOCPMIDLEREQ signal is active), the timer module will go in Sleep mode according to the IdleMode field of the System configuration register (see TIOCP_CFG and Ref. [4]).

If the IdleMode field sets No-Idle mode, the Timer does not go in Sleep mode and the Idle acknowledge signal (POROCPSIDLEACK) is never asserted.

If the IdleMode field sets Force-Idle mode, the timer goes in Sleep mode independently of the internal module state and the Idle acknowledge signal (POROCPSIDLEACK) is unconditionally asserted.

If the IdleMode field sets Smart-Idle mode, the timer module evaluates its internal capability to have the interface/functional clock switched off. Depending on the ClockActivity field setting the timer module evaluates the internal activity and asserts the Idle acknowledge signal (POROCPSIDLEACK), entering in Sleep mode, ready to issue a wake-up request.

The following table describes the Smart Idle behavior according to the clock activity setting:

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 19-31. SmartIdle - Clock Activity Field Configuration

Clock Activity	Functional Clock	OCP Clock	Module Behavior
11	ON	ON	The Idle acknowledge signal is asserted when there are no pending activities on the OCP clock domain, without evaluating the pending activities on the functional clock domain. (The module will enter in Sleep mode and if a pending interrupt event is finished during Idle mode the wake-up signal will be asserted).
10	ON	OFF	
01	OFF	ON	The Idle acknowledge signal is asserted when there are no pending activities on the functional and OCP clock domains (Improved latency in assertion of Idle acknowledge). The Wake-up capability of the module is disabled.
00	OFF	OFF	

This wake-up request is effectively sent only if the field ENAWAKEUP of TIOCP_CFG enables the timer wake-up capability. When the system is awaken, the Idle Request signal goes inactive and the wake-up request signal is also de-asserted.

Figure 19-31. Wake-up Request Generation

19.2.3.7.1 Wake-up Line Release

When the host processor receives wake-up request issued by the timer peripheral, the interface clock is re-activated: the host processor deactivates the PIOCPMIDLEREQ, the timer deactivates the POROCPSIDLEACK signal and then the host can read the corresponding bit in TISR to find out which interrupt source has trigged wake-up request. After acknowledging the wake-up request, the processor resets the status bit and releases the interrupt line by writing a ‘1’ in the corresponding bit of the TISR register.

19.2.3.8 Timer Counting Rate

The dmtimer’s counter is composed of a prescaler stage and a timer counter.

Ratio can be managed by accessing the ratio definition field of the control register (PTV and PRE of TCLR).

The timer rate is defined by:

- The value of the prescaler fields (PRE and PTV of TCLR register)
- The value loaded into the Timer Load Register (TLDR).

Table 19-32. Prescaler Clock Ratios Value

PRE	PTV	Divisor (PS)
0	X	1
1	0	2
1	1	4
1	2	8
1	3	16
1	4	32
1	5	64
1	6	128

Copyright © 2011, Texas Instruments Incorporated

Table 19-32. Prescaler Clock Ratios Value (continued)

PRE	PTV	Divisor (PS)
1	7	256

The timer rate equation is as follows:

(0xFFFF FFFF – TLDR + 1) x timer Clock period x Clock Divider (PS)

With timer Clock period = 1/ timer Clock frequency and PS = 2(PTV + 1).

As example, if we consider a timer clock input of 32 KHz, with a PRE field equals to “0”, the timer output period is:

Table 19-33. Value and Corresponding Interrupt Period

TLDR	Interrupt Period
0x0000 0000	37 h
0xFFFF 0000	2 s
0xFFFF FFF0	500 us
0xFFFF FFFE	62.5 us

19.2.3.9 Dual Mode Timer Under Emulation

During emulation mode (when PINSUSPENDN signal is active), the timer can/cannot continue running according to the value of the EmuFree bit of the timer OCP configuration register (TIOCP_CFG).

If EmuFree is 1, timer execution is not stopped and, regardless of the value of PINSUSPENDN signal, and the interrupt assertion is still generated when overflow is reached.

If EmuFree is 0, counters (prescaler/timer) are frozen and an increment start occurs again as soon as PINSUSPENDN becomes inactive. The asynchronous input pin is internally synchronized on 2 TIMER clock rising edges.

19.2.4 Use Cases

19.2.5 DMTIMER_1MS Registers

Table 19-34 lists the memory-mapped registers for the DMTIMER_1MS. See your device-specific data manual for the memory address of these registers. All other register offset addresses not listed in Table 19-34 should be considered as reserved locations and the register contents should not be modified.

Table 19-34. DMTIMER_1MS REGISTERS

Offset	Acronym	Register Description	Section
0h	TIDR	This register contains the IP revision code	Section 19.2.5.1
10h	TIOCP_CFG	This register controls the various parameters of the OCP interface	Section 19.2.5.2
14h	TISTAT	This register provides status information about the module, excluding the interrupt status information	Section 19.2.5.3
18h	TISR	The Timer Status Register is used to determine which of the timer events requested an interrupt.	Section 19.2.5.4
1Ch	TIER	This register controls (enable/disable) the interrupt events	Section 19.2.5.5
20h	TWER	This register controls (enable/disable) the wakeup feature on specific interrupt events	Section 19.2.5.6
24h	TCLR	This register controls optional features specific to the timer functionality	Section 19.2.5.7
28h	TCRR	This register holds the value of the internal counter	Section 19.2.5.8
2Ch	TLDR	This register holds the timer's load value	Section 19.2.5.9

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Table 19-34. DMTIMER_1MS REGISTERS (continued)

Offset	Acronym	Register Description	Section
30h	TTGR	This register triggers a counter reload of timer by writing any value in it.	Section 19.2.5.10
34h	TWPS	This register contains the write posting bits for all writ-able functional registers	Section 19.2.5.11
38h	TMAR	This register holds the match value to be compared with the counter's value	Section 19.2.5.12
3Ch	TCAR1	This register holds the value of the first counter register capture	Section 19.2.5.13
40h	TSICR	Timer Synchronous Interface Control Register	Section 19.2.5.14
44h	TCAR2	This register holds the value of the second counter register capture	Section 19.2.5.15
48h	TPIR	This register is used for 1ms tick generation. The TPIR register holds the value of the positive increment. The value of this register is added with the value of the TCVR to define whether next value loaded in TCRR will be the sub-period value or the over-period value.	Section 19.2.5.16
4Ch	TNIR	This register is used for 1ms tick generation. The TNIR register holds the value of the negative increment. The value of this register is added with the value of the TCVR to define whether next value loaded in TCRR will be the sub-period value or the over-period value.	Section 19.2.5.17
50h	TCVR	This register is used for 1ms tick generation. The TCVR register defines whether next value loaded in TCRR will be the sub-period value or the over-period value.	Section 19.2.5.18
54h	TOCR	This register is used to mask the tick interrupt for a selected number of ticks.	Section 19.2.5.19
58h	TOWR	This register holds the number of masked overflow interrupts.	Section 19.2.5.20

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.2.5.1 TIDR Register (offset = 0h) [reset = 0h]

This register contains the IP revision code

TIDR is shown in Figure 19-32 and described in Table 19-35.

Figure 19-32. TIDR Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 19-35. TIDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-8	Reserved	R	0h	Reads return 0
7-0	TID_REV	R	0h	IP revision [7:4] Major revision [3:0] Minor revision Examples: 0x10 for 1.0, 0x21 for 2.1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.2.5.2 TIOCP_CFG Register (offset = 10h) [reset = 0h]

This register controls the various parameters of the OCP interface

TIOCP_CFG is shown in Figure 19-33 and described in Table 19-36.

Figure 19-33. TIOCP_CFG Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 19-36. TIOCP_CFG Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	Reserved	R/W	0h	
9-8	ClockActivity	R/W	0h	0 - CA0: Functional clock can be switched-off OCP clock can be switched-off 1 - CA1: Functional clock can be switched-off OCP clock is maintained during wake-up period 2 - CA2: Functional clock is maintained during wake-up period OCP clock can be switched-off 3 - CA3: Functional clock is maintained during wake-up period OCP clock is maintained during wake-up period
7-6	Reserved	R/W	0h	Write 0's for future compatibility Reads return 0
5	EmuFree	R/W	0h	Emulation mode 0 - timer_frozen: Timer counter frozen in emulation 1 - timer_free: Timer counter free-running in emulation
4-3	IdleMode	R/W	0h	Power Management, req/ack control 0 - fidle: Force-idle. An idle request is acknowledged unconditionally 1 - nidle: No-idle. An idle request is never acknowledged 2 - sidle: Smart-idle. Acknowledgement to an idle request is given based on the internal activity of the module 3 - res: reserved do not use
2	EnaWakeup	R/W	0h	Wake-up feature global control 0 - nowake: No wakeup line assertion in idle mode 1 - enbwake: Wakeup line assertion enabled in smart-idle mode
1	SoftReset	R/W	0h	Software reset. This bit is automatically reset by the hardware. During reads, it always return 0 0 - nmode: Normal mode 1 - rstmode: The module is reset
0	AutoIdle	R/W	0h	Internal OCP clock gating strategy 0 - clkfree: OCP clock is free-running 1 - clkgate: Automatic OCP clock gating strategy is applied, based on the OCP interface activity

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.2.5.3 TISTAT Register (offset = 14h) [reset = 0h]

This register provides status information about the module, excluding the interrupt status information TISTAT is shown in Figure 19-34 and described in Table 19-37.

Figure 19-34. TISTAT Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 19-37. TISTAT Register Field Descriptions

Bit	Field	Type	Reset	Description
31-1	Reserved	R	0h	Reads return 0 Reserved for OCP-socket status information
0	ResetDone	R	0h	Internal reset monitoring 0 - rstongoing: Internal module reset in on-going 1 - rstcomp: Reset completed

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.2.5.4 TISR Register (offset = 18h) [reset = 0h]

The Timer Status Register is used to determine which of the timer events requested an interrupt.

TISR is shown in Figure 19-35 and described in Table 19-38.

Figure 19-35. TISR Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 19-38. TISR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	Reserved	R	0h	Reads return 0
2	TCAR_IT_FLAG	R/W	0h	indicates when an external pulse transition of the correct polarity is detected on the external pin PIEVENTCAPT 0 - TCAR_IT_FLAG_0: no capture interrupt request 1 - TACR_IT_FLAG_1: capture interrupt request
1	OVF_IT_FLAG	R/W	0h	TCRR overflow 0 - OVF_IT_FLAG_0: no overflow interrupt request 1 - OVF_IT_FLAG_1: overflow interrupt pending
0	MAT_IT_FLAG	R/W	0h	the compare result of TCRR and TMAR 0 - MAT_IT_FLAG_0: no compare interrupt request 1 - MAT_IT_FLAG_1: compare interrupt pending

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.2.5.5 TIER Register (offset = 1Ch) [reset = 0h]

This register controls (enable/disable) the interrupt events

TIER is shown in Figure 19-36 and described in Table 19-39.

Figure 19-36. TIER Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 19-39. TIER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	Reserved	R	0h	Reads return 0
2	TCAR_IT_ENA	R/W	0h	Enable capture interrupt 0 - Dsb_capt: Disable capture interrupt 1 - Enb_capt: Enable capture interrupt
1	OVF_IT_ENA	R/W	0h	Enable overflow interrupt 0 - Dsb_ovf: Disable overflow interrupt 1 - Enb_ovf: Enable overflow interrupt
0	MAT_IT_ENA	R/W	0h	Enable match interrupt 0 - Dsb_match: Disable match interrupt 1 - Enb_match: Enable match interrupt

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.2.5.6 TWER Register (offset = 20h) [reset = 0h]

This register controls (enable/disable) the wakeup feature on specific interrupt events

TWER is shown in Figure 19-37 and described in Table 19-40.

Figure 19-37. TWER Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 19-40. TWER Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	Reserved	R	0h	Reads return 0
2	TCAR_WUP_ENA	R/W	0h	Enable capture wake-up 0 - DsbWupCap: Disable capture wake-up 1 - EnbWupCapt: Enable capture wake-up
1	OVF_WUP_ENA	R/W	0h	Enable overflow wake-up 0 - DsbWupOvf: Disable overflow wake-up 1 - EnbWupOvf: Enable overflow wake-up
0	MAT_WUP_ENA	R/W	0h	Enable match wake-up 0 - DsbWupMat: Disable match wake-up 1 - EnbWupMat: Enable match wake-up

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.2.5.7 TCLR Register (offset = 24h) [reset = 0h]

This register controls optional features specific to the timer functionality

TCLR is shown in Figure 19-38 and described in Table 19-41.

Figure 19-38. TCLR Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 19-41. TCLR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-15	Reserved	R	0h	Reads return 0
14	GPO_CFG	R/W	0h	0 - GPO_CFG_0: PORGPOCFG drives 0 1 - GPO_CFG_1: PORGPOCFG drives 1
13	CAPT_MODE	R/W	0h	Capture mode select bit (first/second) 0 - First_capt: Capture the first enabled capture event in TCAR1 1 - Sec_capt: Capture the second enabled capture event in TCAR2
12	PT	R/W	0h	Pulse or Toggle select bit 0 - pulse: pulse modulation 1 - toggle: toggle modulation
11-10	TRG	R/W	0h	Trigger Output Mode 0 - no_trg: No trigger 1 - ovf_trg: Overflow trigger 2 - ovf_mat_trg: Overflow and match trigger 3 - reserved: Reserved
9-8	TCM	R/W	0h	Transition Capture Mode 0 - no_edge: No capture 1 - rise_edge: Capture on rising edges of PIEVETCAPT 2 - fall_edge: Capture on falling edges of PIEVETCAPT 3 - booth_edges: Capture on booth edges of PIEVETCAPT
7	SCPWM	R/W	0h	Pulse Width Modulation output pin default value 0 - def_low: default value of PORPWM: 0 1 - def_high: default value of PORPWM: 1
6	CE	R/W	0h	Compare enable 0 - dsb_cmp: Compare disabled 1 - enb_cmp: Compare enabled
5	PRE	R/W	0h	Prescaler enable 0 - no_prescal: Prescaler disabled 1 - prescal_on: Prescaler enabled
4-2	PTV	R/W	0h	Trigger Output Mode 0 - ptv: The timer counter is prescaled with the value: 2PTV. Example: PTV = 3: counter increases value is started after 8 functional clock periods
1	AR	R/W	0h	Auto-reload mode 0 - one_shot: One shot mode overflow 1 - auto_rel: Auto-reload mode overflow
0	ST	R/W	0h	Start/Stop timer control 0 - cnt_stop: Stop the timer 1 - cnt_start: Start the timer

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.2.5.8 TCRR Register (offset = 28h) [reset = 0h]

This register holds the value of the internal counter

TCRR is shown in Figure 19-39 and described in Table 19-42.

Figure 19-39. TCRR Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 19-42. TCRR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TIMER_COUNTER	R/W	0h	The value of the timer counter register

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.2.5.9 TLDR Register (offset = 2Ch) [reset = 0h]

This register holds the timer's load value

TLDR is shown in Figure 19-40 and described in Table 19-43.

Figure 19-40. TLDR Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 19-43. TLDR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	LOAD_VALUE	R/W	0h	The value of the timer load register

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.2.5.10 TTGR Register (offset = 30h) [reset = FFFFFFFFh]

This register triggers a counter reload of timer by writing any value in it.

TTGR is shown in Figure 19-41 and described in Table 19-44.

Figure 19-41. TTGR Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 19-44. TTGR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	TTGR_VALUE	R/W	FFFFFFFFh	The value of the trigger register During reads, it always returns "0xFFFFFFFF"

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.2.5.11 TWPS Register (offset = 34h) [reset = 0h]

This register contains the write posting bits for all writ-able functional registers

TWPS is shown in Figure 19-42 and described in Table 19-45.

Figure 19-42. TWPS Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 19-45. TWPS Register Field Descriptions

Bit	Field	Type	Reset	Description
31-10	Reserved	R	0h	Reads return 0
9	W_PEND_TOWR	R	0h	Write pending for register TOWR 0 - OWR_nPend: No Overflow Wrapping Register write pending. 1 - OWR_Pend: Overflow Wrapping Register write pending.
8	W_PEND_TOCR	R	0h	Write pending for register TOCR 0 - OCR_nPend: No Overflow Counter Register write pending. 1 - OCR_Pend: Overflow Counter Register write pending.
7	W_PEND_TCVR	R	0h	Write pending for register TCVR 0 - CVR_nPend: No Counter Register write pending. 1 - CVR_Pend: Counter Register write pending.
6	W_PEND_TNIR	R	0h	Write pending for register TNIR 0 - NIR_nPend: No Negativ Increment Register write pending. 1 - NIR_Pend: Negativ Increment Register write pending.
5	W_PEND_TPIR	R	0h	Write pending for register TPIR 0 - PIR_nPend: No Positive Increment Register write pending. 1 - PIR_Pend: Positive Increment Register write pending.
4	W_PEND_TMAR	R	0h	Write pending for register TMAR 0 - MAR_nPend: No Match Register write pending 1 - MAR_Pend: Match Register write pending
3	W_PEND_TTGR	R	0h	Write pending for register TTGR 0 - TGR_nPend: No Trigger Register write pending 1 - TGR_Pend: Trigger Register write pending
2	W_PEND_TLDR	R	0h	Write pending for register TLDR 0 - LDR_nPend: No Load Register write pending 1 - LDR_Pend: Load Register write pending
1	W_PEND_TCRR	R	0h	Write pending for register TCRR 0 - CRR_nPend: No Counter Register write pending 1 - CRR_Pend: Counter Register write pending
0	W_PEND_TCLR	R	0h	Write pending for register TCLR 0 - CLR_nPend: No Control Register write pending 1 - CLR_Pend: Control Register write pending

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.2.5.12 TMAR Register (offset = 38h) [reset = 0h]

This register holds the match value to be compared with the counter's value

TMAR is shown in Figure 19-43 and described in Table 19-46.

Figure 19-43. TMAR Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 19-46. TMAR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COMPARE_VALUE	R/W	0h	The value of the match register

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.2.5.13 TCAR1 Register (offset = 3Ch) [reset = 0h]

This register holds the value of the first counter register capture

TCAR1 is shown in Figure 19-44 and described in Table 19-47.

Figure 19-44. TCAR1 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 19-47. TCAR1 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAPTURE_VALUE1	R	0h	The value of first captured counter register

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.2.5.14 TSICR Register (offset = 40h) [reset = 0h]

Timer Synchronous Interface Control Register

TSICR is shown in Figure 19-45 and described in Table 19-48.

Figure 19-45. TSICR Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 19-48. TSICR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-3	Reserved	R	0h	Reads return 0
2	POSTED	R/W	0h	posted mode active/inactive 0 - POSTED_0: posted mode inactive: will delay the command accept output signal 1 - POSTED_1: posted mode active (clocks ratio needs to fit freq (timer) andlt freq (OCP)/4 frequency requirement)
1	SFT	R/W	0h	This bit reset all the functional part of the module 0 - SFT_0: software reset is disabled 1 - SFT_1: software reset is enabled
0	Reserved	R	0h	Reads return 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.2.5.15 TCAR2 Register (offset = 44h) [reset = 0h]

This register holds the value of the second counter register capture

TCAR2 is shown in Figure 19-46 and described in Table 19-49.

Figure 19-46. TCAR2 Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 19-49. TCAR2 Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	CAPTURE_VALUE2	R	0h	The value of second captured counter register

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.2.5.16 TPIR Register (offset = 48h) [reset = 0h]

This register is used for 1ms tick generation. The TPIR register holds the value of the positive increment. The value of this register is added with the value of the TCVR to define whether next value loaded in TCRR will be the sub-period value or the over-period value.

TPIR is shown in Figure 19-47 and described in Table 19-50.

Figure 19-47. TPIR Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 19-50. TPIR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	POSITIVE_INC_VALUE	R/W	0h	The value of the positive increment.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.2.5.17 TNIR Register (offset = 4Ch) [reset = 0h]

This register is used for 1ms tick generation. The TNIR register holds the value of the negative increment. The value of this register is added with the value of the TCVR to define whether next value loaded in TCRR will be the sub-period value or the over-period value.

TNIR is shown in Figure 19-48 and described in Table 19-51.

Figure 19-48. TNIR Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 19-51. TNIR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	NEGATIVE_INV_VALUE	R/W	0h	The value of the negative increment.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.2.5.18 TCVR Register (offset = 50h) [reset = 0h]

This register is used for 1ms tick generation. The TCVR register defines whether next value loaded in TCRR will be the sub-period value or the over-period value.

TCVR is shown in Figure 19-49 and described in Table 19-52.

Figure 19-49. TCVR Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -*n* = value after reset

Table 19-52. TCVR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-0	COUNTER_VALUE	R/W	0h	The value of CVR counter.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.2.5.19 TOCR Register (offset = 54h) [reset = 0h]

This register is used to mask the tick interrupt for a selected number of ticks.

TOCR is shown in Figure 19-50 and described in Table 19-53.

Figure 19-50. TOCR Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 19-53. TOCR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	Reserved	R	0h	Reads return 0.
23-0	OVF_COUNTER_VALUE	R/W	0h	The number of overflow events.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.2.5.20 TOWR Register (offset = 58h) [reset = 0h]

This register holds the number of masked overflow interrupts.

TOWR is shown in Figure 19-51 and described in Table 19-54.

Figure 19-51. TOWR Register

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; *-n* = value after reset

Table 19-54. TOWR Register Field Descriptions

Bit	Field	Type	Reset	Description
31-24	Reserved	R	0h	Reads return 0
23-0	OVF_WRAPPING_VALUE	R/W	0h	The number of masked interrupts.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.3 RTC_SS

19.3.1 Introduction

The real-time clock is a precise timer which can generate interrupts on intervals specified by the user. Interrupts can occur every second, minute, hour, or day. The clock itself can track the passage of real time for durations of several years, provided it has a sufficient power source the whole time.

The basic purpose for the RTC is to keep time of day. The other equally important purpose of RTC is for Digital Rights management. Some degree of tamper proofing is needed to ensure that simply stopping, resetting, or corrupting the RTC does not go unnoticed so that if this occurs, the application can re-acquire the time of day from a trusted source. The final purpose of RTC is to wake the rest of chip up from a power down state.

Alarms are available to interrupt the CPU at a particular time, or at periodic time intervals, such as once per minute or once per day. In addition, the RTC can interrupt the CPU every time the calendar and time registers are updated, or at programmable periodic intervals.

19.3.1.1 Features

The real-time clock (RTC) provides the following features:

- 100-year calendar (xx00 to xx99)
- Counts seconds, minutes, hours, day of the week, date, month, and year with leap year compensation
- Binary-coded-decimal (BCD) representation of time, calendar, and alarm
- 12-hour clock mode (with AM and PM) or 24-hour clock mode
- Alarm interrupt
- Periodic interrupt
- Single interrupt to the CPU
- Supports external 32.768-kHz crystal or external clock source of the same frequency

19.3.1.2 Unsupported RTC Features

This device supports only a single RTC external wake-up event.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.3.2 Integration

This device includes a Real-Time Clock Subsystem (RTCSS) module to allow easy tracking of time and date and the generation of real time alarms.

19.3.2.1 RTC Connectivity Attributes

The general connectivity for the RTC module in the device is shown in Table 19-55.

Table 19-55. RTC Module Connectivity Attributes

Attributes	Type
Power Domain	RTC
Clock Domain	PD_RTC_L4_RTC_GCLK (OCP) PD_RTC_RTC32KCLK (Func) CLK_32K_RTC (Func)
Reset Signals	RTC_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	4 Interrupts Alarm interrupt to Sabertooth & WakeM3 Timer interrupt to Sabertooth and WakeM3 Alarm wakeup to WakeM3 Timer wakeup to WakeM3
DMA Requests	None
Physical Address	L4 Wakeup slave port

19.3.2.2 RTC Clock and Reset Management

The RTC functional clock (c32khz_clk input) is sourced by default from the CLK32_KHZ clock derived from the Peripheral PLL. It can also be sourced from the 32-KHz oscillator through a clock mux within and controlled by the RTC_SS.

Table 19-56. RTC Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
ocp_clk (Interface clock)	100 MHz	CORE_CLKOUTM4 / 2	pd_rtc_l4_rtc_gclk From PRCM
rtc_32K_clk_rtc_32k_clk (Oscillator functional clock)	32.768 KHz	OSC1 In	CLK_32K_RTC From OSC1 In
rtc_32k_clk_rtc_32k_aux_clk (Internal functional clock)	32.768 KHz	PER_CLKOUTM2 / 5859.3752	pd_rtc_rtc_32kclk From PRCM

19.3.2.3 RTC Pin List

The RTC module does not include any external interface pins.

Table 19-57. RTC Pin List

Pin	Type	Description
RTC_PORz	I	RTC Power On Reset
EXT_WAKEUP[3:0]	I	External wakeup
PMIC_POWER_EN	O	Power enable control for external power management IC
Analog Signals		
ENZ_KALDO_1P8V	I	Enable1.8V LDO
VDDS18_RTC	P	1.8V Voltage Supply

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.3.3 Functional Description

This section defines the module interrupt capabilities and requirements.

19.3.3.1 Functional Block Diagram

Figure 19-52 shows the RTC module block diagram. Figure 19-53 shows a functional block diagram of the RTC.

Figure 19-52. RTC Block Diagram

Figure 19-53. RTC Functional Block Diagram

19.3.3.2 Clock Source

The clock reference for the RTC is an internal 32.768-kHz crystal or an external clock source of the same frequency. The source for the RTC reference clock may be provided by a crystal or by an external clock source. The RTC has an internal oscillator buffer to support direct operation with a crystal. The crystal is connected between pins RTC_XI and RTC_XO. RTC_XI is the input to the on-chip oscillator and RTC_XO is the output from the oscillator back to the crystal. For more information about the RTC crystal connection, see your device-specific data manual.

An external 32.768-kHz clock source may be used instead of a crystal. In such a case, the clock source is connected to RTC_XI, and RTC_XO is left unconnected.

If the RTC is not used, the RTC_XI pin should be held low and RTC_XO should be left unconnected. The RTC_disable bit in the control register (CTRL_REG) can be set to save power; however, the RTC_disable bit should not be cleared once it has been set. If the application requires the RTC module to stop and continue, the STOP_RTC bit in CTRL should be used instead.

19.3.3.3 Signal Descriptions

Table 19-58 lists the signals and their descriptions for the RTC.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 19-58. RTC Signals

Signal	I/O	Description
RTC_XI	I	RTC time base input signal. RTC_XI can either be driven with a 32.768-kHz reference clock, or RTC_XI and RTC_XO can be connected to an external crystal. This signal is the input to the RTC internal oscillator.
RTC_XO	O	RTC time base output signal. RTC_XO is the output from the RTC internal oscillator. If a crystal is not used as the time base for RTC_XI, RTC_XO should be left unconnected.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.3.3.4 Interrupt Support

19.3.3.4.1 CPU Interrupts

The RTC generates two interrupt outputs:

- timer_intr is a timer interrupt.
- alarm_intr is an alarm interrupt.

NOTE: Both interrupt outputs support high-level and high-pulse.

19.3.3.4.2 Interrupt Description

19.3.3.4.2.1 Timer Interrupt (timer_intr)

The timer interrupt can be generated periodically: every second, every minute, every hour, or every day (see INTERRUPTS_REG[1:0] for a description of how to set this up). The IT_TIMER bit of the interrupt register enables this interrupt. The timer interrupt is active-low.

The RTC_STATUS_REG[5:2] are only updated at each new interrupt and occur according to Table 19-59. For example, bit 2 (SEC) will always be set when one second has passed. It will also be set when one minute has passed since the completion of one minute also marks the completion of one second (from 59 seconds to 60 seconds). The same holds true for hours and days: each of them will also correspond to the passing of a second.

Conversely, bit 5 (DAY) will always be set when a day has passed. It might also be set when an hour, minute, or second has passed. However, this only occurs when the elapsed hour, minute, or second corresponds to the start of a new day.

Table 19-59. Interrupt Trigger Events

	One day has passed	One hour has passed	One minute has passed	One second has passed
STATUS_REG[5] (DAY)	1	0/1 (1)	0/1 (1)	0/1 (1)
STATUS_REG[4] (HOUR)	1	1	0/1 (1)	0/1 (1)
STATUS_REG[3] (MIN)	1	1	1	0/1 (1)
STATUS_REG[2] (SEC)	1	1	1	1

(1) This event is only triggered when the elapsed time unit (for example, Day) corresponds to the passage of another unit (for example, Seconds). For example, when the clock ticks from 00:23:59:59 (days : hours : minutes : seconds) to 01:00:00:00.

19.3.3.4.2.2 Alarm Interrupt (alarm_intr)

The alarm interrupt can be generated when the time set into TC ALARM registers is exactly the same as in the TC registers. This interrupt is then generated if the IT_ALARM bit of the interrupts register is set. This interrupt is low-level sensitive. RTC_STATUS_REG[6] indicates that IRQ_ALARM_CHIP has occurred. This interrupt is disabled by writing ‘1’ into the RTC_STATUS_REG[6].

To set up an alarm:

- Modify the ALARM_SECONDS, ALARM_MINUTES, ALARM_HOURS, ALARM_DAY, ALARM_WEEK, ALARM_MONTH, and ALARM_YEAR registers to the exact time you want an alarm to generate.
- Set the IT_ALARM bit in the RTC_INTERRUPTS register to enable the alarm interrupt.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.3.3.5 Programming/Usage Guide

19.3.3.5.1 Time/Calendar Data Format

The time and calendar data in the RTC is stored as binary-coded decimal (BCD) format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. Although most of the time/calendar registers have 4 bits assigned to each BCD digit, some of the register fields are shorter since the range of valid numbers may be limited. For example, only 3 bits are required to represent the day of the week (WEEKS_REG) since only BCD numbers 1 through 7 are required. The following time and calendar registers are supported (BCD Format):

Note that the ALARM registers which share the names above also share the same BCD formatting.

- SECOND - Second Count (00-59)
- MINUTE - Minute Count (00-59)
- HOUR - Hour Count (12HR: 01-12; 24HR: 00-23)
- DAY - Day of the Month Count (01-31)
- WEEK - Day of the Week (0-6: SUN = 0)
- MONTH - Month Count (01-12; JAN = 1)
- YEAR - Year Count (00-99)

19.3.3.5.2 Register Access

The three register types are as follows and each has its own access constraints:

- TC and TC alarm registers
- General registers
- Compensation registers

19.3.3.5.3 OCP MMR Spurious WRT Protection

The module also contains a kicker mechanism (Figure 19-54) to prevent any spurious writes from changing the register values. This mechanism requires two MMR writes to the Kick0 and Kick1 registers with exact data values before the kicker lock mechanism is released. Once released, the MMRs are writeable. The Kick0 data is 83E7 0B13h; the Kick1 data is 95A4 F1E0h. Note that it remains in an unlocked state until an OCP reset or invalid data pattern is written to one of the Kick0 or Kick1 registers.

Figure 19-54. Kick Register State Machine Diagram

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

- S0 is the Reset/Idle state
- S1 is an OCP wrt cycle of 83E7 0B13h at Kick0 completed state
- S2 is the UNLOCK MMR WRT state
- S0 -> S1 when OCP wrt cycle of 83E7 0B13h at Kick0
- S1 -> S2 when OCP wrt cycle of 95A4 F1E0h at Kick1
- S1 -> S0 when OCP reset event
- S2 -> S0 when OCP reset event OR OCP wrt cycle of NOT 83E7 0B13h at Kick0 OR OCP wrt cycle at Kick1
- S2 ->S1 when OCP wrt cycle of 83E7 0B13h at Kick0

19.3.3.5.4 Reading the Timer/Calendar (TC) Registers

The TC registers have a read-show register. The reading of the Seconds register will update all of the TC registers. For example, the Year will only get updated on a reading of the Seconds register. The time/calendar registers are updated every second as the time changes. During a read of the SECONDS register, the RTC copies the current values of the time/date registers into shadow read registers. This isolation assures that the CPU can capture all the time/date values at the moment of the SECONDS read request and not be subject to changing register values from time updates.

If desired, the RTC also provides a one-time-triggered minute-rounding feature to round the MINUTE:SECOND registers to the nearest minute (with zero seconds). This feature is enabled by setting the ROUND_30S bit in the control register (CTRL); the RTC automatically rounds the time values to the nearest minute upon the next read of the SECONDS register.

NOTE: Software should always read the Seconds register first. However, the software does not have to poll any status bit to determine when to read the TC registers. Table 19-60 defines the TC set that gets shadowed.

Table 19-60. RTC Register Names and Values

Time Unit	Range	Remarks
Year	00 to 99	
Month	01 to 12	
Day	01 to 31	Months 1, 3, 5, 7, 8, 10, 12
	01 to 30	Months 4, 6, 9, 11
	01 to 29	Month 2 (leap year)
	01 to 28	Month 2 (common year)
Week	00 to 06	Day of week
Hour	00 to 23	24 hour mode
	01 to 12	AM/PM mode
Minute	00 to 59	
Seconds	00 to 59	

19.3.3.5.4.1 Rounding Seconds

Time can be rounded to the closest minute, by setting the ROUND_30S bit of the control register. When this bit is set, TC values are set to the closest minute value at the next second. The ROUND_30S bit will be automatically cleared when rounding time is performed.

Example:

- If current time is 10H59M45S, round operation will change time to 11H00M00S.
- If current time is 10H59M29S, round operation will change time to 10H59M00S.

Copyright © 2011, Texas Instruments Incorporated

19.3.3.5.5 Modifying the TC Registers

To write correct data from/to the TC and TC alarm registers and read the TC alarm registers, the ARM must first read the BUSY bit of the STATUS register until BUSY is equal to zero. Once the BUSY flag is zero, there is a 15 µs access period in which the ARM can program the TC and TC alarm registers. Once the 15 µs access period passes, the BUSY flag has to be read again from the STATUS register as described previously. If the ARM accesses the TC registers outside of the access period, then the access is not guaranteed.

The ARM can access the STATUS_REG and CTRL_REG at any time, with the exception of CTRL_REG[5] which can only be changed when the RTC is stopped. The ARM can stop the RTC by clearing the STOP_RTC bit of the control register. After clearing this bit, the RUN bit in the STATUS_REG (bit 1) needs to be checked to verify the RTC has in fact stopped. Once this is confirmed, the TC values can be updated. After the values have been updated, the RTC can be re-started by resetting the STOP_RTC bit.

NOTE: After writing to a TC register, the user must wait 4 OCP clock cycles before reading the value from the register. If this wait time is not observed and the TC register is accessed, then old data will be read from the register.

CAUTION

In order to remove any possibility of interrupting the register's read process, thus introducing a potential risk of violating the authorized 15-microsecond access period, it is strongly recommended that you disable all incoming interrupts during the register read process.

Figure 19-55. Flow Control for Updating RTC Registers

19.3.3.5.5.1 General Registers

The ARM can access the STATUS_REG and the CTRL_REG at any time (except the CTRL_REG[5] bit which must be changed only when the RTC is stopped). For the INTERRUPTS_REG, the ARM should respect the available access period to prevent spurious interrupt.

Copyright © 2011, Texas Instruments Incorporated

The RTC_DISABLE bit of the CTRL register must only be used to completely disable the RTC function. When this bit is set, the 32 kHz clock is gated, and the RTC is frozen. From this point, resetting this bit to zero can lead to unexpected behavior. In order to save power, this bit should only be used if the RTC function is unwanted in the application.

19.3.3.5.6 Crystal Compensation

To compensate for any inaccuracy of the 32 kHz oscillator, the ARM can perform a calibration of the oscillator frequency, calculate the drift compensation versus one-hour period, and load the compensation registers with the drift compensation value. Auto compensation is enabled by AUTO_COMP_EN bit in the RTC_CTRL register. If the COMP_REG value is positive, compensation occurs after the second change event. COMP_REG cycles are removed from the next second. If the COMP_REG value is negative, compensation occurs before the second change event. COMP_REG cycles are added to the current second. This enables compensation with a 1 32-kHz period accuracy each hour. The waveform below summarizes positive and negative compensation effect.

Access to the COMP_MSB_REG and COMP_LSB_REG registers must respect the available access period. These registers should not be updated during compensation (first second of each hour), but it is alright to update them during the second preceding a compensation event. For example, the ARM could load the compensation value into these registers after each hour event, during an available access period.

Figure 19-56. Compensation Illustration

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.3.3.6 Scratch Registers

The RTC provides three general-purpose registers (SCRATCHx_REG) that can be used to store 32-bit words -- these registers have no functional purpose for the RTC. Software using the RTC may find the SCRATCHx registers to be useful in indicating RTC states. For example, the SCRATCHx_REG registers may be used to indicate write-protection lock status or unintentional power downs. To indicate write-protection, the software should write a unique value to one of the SCRATCHx_REG registers when write-protection is disabled and another unique value when write-protection is enabled again. In this way, the lock-status of the registers can be determined quickly by reading the SCRATCH register. To indicate unintentional power downs, the software should write a unique value to one of the SCRATCHx_REG registers when RTC is configured and enabled. If the RTC is unintentionally powered down, the value written to the SCRATCH register is cleared. For more information, see the registers section.

19.3.3.7 Power Management

The RTC supports the power idle protocol. It has two SWakeup ports: one for the alarm event and one for a timer event.

When the RTC is in IDLE mode, the OCP clock is turned off and the 32 kHz clock remains on. The time and calendar continue to count in IDLE mode. When the RTC is placed back in FUNCTIONAL mode, the TC registers can be read.

The Alarm SWakeup event can be used to wakeup the RTC when it is in IDLE state. In order to do so, the alarm needs to be set and enabled before RTC enters the IDLE state. Once this is done, the SWakeup will occur when the alarm event triggers.

NOTE: Since SWakeup is not periodic, using it to wake up the RTC when in IDLE state is not recommended. Please use Alarm SWakeup instead.

19.3.3.8 Reset Considerations

When the device is initially powered on, the RTC may issue spurious interrupt signals to the CPU. To avoid issues, a software reset should be performed on the RTC module before the CPU interrupt controller is initialized. Also, see the register summary table in the registers section for reset states of each register.

19.3.4 Use Cases

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.3.5 RTC Registers

Table 19-61 lists the memory-mapped registers for the RTC. See your device-specific data manual for the memory address of these registers.

Table 19-61. Real-Time Clock (RTC) Registers

Address Offset	Acronym	Register Description
0h	SECONDS_REG	Seconds Register
4h	MINUTES_REG	Minutes Register
8h	HOURS_REG	Hours Register
Ch	DAYS_REG	Day of the Month Register
10h	MONTHS_REG	Month Register
14h	YEARS_REG	Year Register
18h	WEEK_REG	Day of the Week Register
20h	ALARM_SECONDS_REG	Alarm Seconds Register
24h	ALARM_MINUTES_REG	Alarm Minutes Register
28h	ALARM_HOURS_REG	Alarm Hours Register
2Ch	ALARM_DAYS_REG	Alarm Days Register
30h	ALARM_MONTHS_REG	Alarm Months Register
34h	ALARM_YEARS_REG	Alarm Years Register
40h	RTC_CTRL_REG	Control Register
44h	RTC_STATUS_REG	Status Register
48h	RTC_INTERRUPTS_REG	Interrupt Enable Register
4Ch	RTC_COMP_LSB_REG	Compensation (LSB) Register
50h	RTC_COMP_MSB_REG	Compensation (MSB) Register
54h	RTC_OSC_REG	Oscillator Register
60h	RTC_SCRATCH0_REG	Scratch 0 Register (General-Purpose)
64h	RTC_SCRATCH1_REG	Scratch 1 Register (General-Purpose)
68h	RTC_SCRATCH2_REG	Scratch 2 Register (General-Purpose)
6Ch	KICK0R	Kick 0 Register (Write Protect)
70h	KICK1R	Kick 1 Register (Write Protect)
74h	RTC_REVISION	Revision Register
78h	RTC_SYSCONFIG	System Configuration Register
7Ah	RTC_IRQWAKEEN_0	Wakeup Enable Register

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.3.5.1 Seconds Register (SECONDS_REG)

The SECONDS_REG is used to program the required seconds value of the current time. Seconds are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. If the seconds value is 45, then the value of SEC0 is 5 and value of SEC1 is 4.

Figure 19-57. Seconds Register (SECONDS_REG)

31			16
Reserved			
R-0			

15	7	6	4	3	0
Reserved		SEC1		SEC0	
R-0		R/W-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-62. Seconds Register (SECONDS_REG) Field Descriptions

Bit	Field	Value	Description
31-7	Reserved	0	Reserved.
6-4	SEC1	0-5h	2nd digit of seconds, Range is 0 to 5
3-0	SEC0	0-9h	1st digit of seconds, Range is 0 to 9

19.3.5.2 Minutes Register (MINUTES_REG)

The MINUTES_REG is used to program the minutes value of the current time. Minutes are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. If the minutes value is 32, then the value of MIN0 is 2 and value of MIN1 is 3.

Figure 19-58. Minutes Register (MINUTES_REG)

31			16
Reserved			
R-0			

15	7	6	4	3	0
Reserved		MIN1		MIN0	
R-0		R/W-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-63. Minutes Register (MINUTES_REG) Field Descriptions

Bit	Field	Value	Description
31-7	Reserved	0	Reserved
6-4	MIN1	0-5h	2nd digit of minutes, Range is 0 to 5
3-0	MIN0	0-9h	1st digit of minutes, Range is 0 to 9

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.3.5.3 Hours Register (HOURS_REG)

The HOURS_REG is used to program the hours value of the current time. Hours are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. In 24Hr time mode if you want to set the hour as 18, then HOUR0 is set as 8 and HOUR1 is set as 1.

Figure 19-59. Hours Register (HOURS_REG)

31							16
Reserved							
R-0							

15	8	7	6	5	4	3	0
Reserved		PM_nAM	Rsvd	HOUR1		HOUR0	
R-0		R/W-0	R-0	R/W-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-64. Hours Register (HOURS_REG) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7	PM_nAM		Only used in PM_AM mode (otherwise 0)
		0	AM
		1	PM
6	Reserved	0	Reserved
5-4	HOUR1	0-2h	2nd digit of hours, Range is 0 to 2
3-0	HOUR0	0-9h	1st digit of hours, Range is 0 to 9

19.3.5.4 Days of the Month Register (DAYS_REG)

The DAYS_REG is used to program the day of the month value of the current date. Days are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. If the day value of the date is 28, DAY0 is set as 8 and DAY1 is set as 2.

Figure 19-60. Days of the Month Register (DAYS_REG)

31							16
Reserved							
R-0							

15	6	5	4	3	0
Reserved		DAY1		DAY0	
R-0		R/W-0		R/W-1	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-65. Day of the Month (DAYS_REG) Field Descriptions

Bit	Field	Value	Description
31-6	Reserved	0	Reserved
5-4	DAY1	0-3h	2nd digit of days, Range is 0 to 3
3-0	DAY0	0-9h	1st digit of days, Range is 0 to 9

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.3.5.5 Month Register (MONTHS_REG)

The MONTHS_REG is used to set the month in the year value of the current date. Months are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.

NOTE: Usual notation is taken for month value:

- 01 => January
- 02 => February
- ...
- 12 => December

Figure 19-61. Month Register (MONTHS_REG)

31															16
Reserved															
R-0															

15							5	4	3				0
Reserved								MONTH1	MONTH0				
R-0								R/W-0	R/W-1				

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-66. Month Register (MONTHS_REG) Field Descriptions

Bit	Field	Value	Description
31-5	Reserved	0	Reserved
4	MONTH1	0-1h	2nd digit of months, Range is 0 to 1
3-0	MONTH0	0-9h	1st digit of months, Range is 0 to 9

19.3.5.6 Year Register (YEARS_REG)

The YEARS_REG is used to program the year value of the current date. The year value is represented by only the last 2 digits and is stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent. The year 1979 is programmed as 79 with YEAR0 set as 9 and YEAR1 set as 7.

Figure 19-62. Year Register (YEARS_REG)

31															16
Reserved															
R-0															

15								8	7			4	3				0
Reserved									YEAR1				YEAR0				
R-0									R/W-0				R/W-0				

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-67. Year Register (YEARS_REG) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-4	YEAR1	0-9h	2nd digit of years, Range is 0 to 9
3-0	YEAR0	0-9h	1st digit of years, Range is 0 to 9

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.3.5.7 Day of the Week Register (WEEKS_REG)

The WEEKS_REG is used to program the day of the week value of the current date. The day of the week is stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.

NOTE: Sunday is treated as 0, Monday 1, and ending at Saturday with 6.

Figure 19-63. Day of the Week Register (WEEKS_REG)

31															16
Reserved															
R-0															

15												3	2		0
Reserved													WEEK		
R-0													R/W-0		

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-68. Day of the Week (WEEKS_REG) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Reserved
2-0	WEEK	0-6h	1st digit of days in a week, Range from 0 (Sunday) to 6 (Saturday)

19.3.5.8 Alarm Second Register (ALARM_SECONDS_REG)

The ALARM_SECONDS_REG is used to program the second value for the alarm interrupt. Seconds are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.

Figure 19-64. Alarm Second Register (ALARM_SECONDS_REG)

31															16
Reserved															
R-0															

15							7	6		4	3				0
Reserved								ALARMSEC1			ALARMSEC0				
R-0								R/W-0			R/W-0				

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-69. Alarm Second Register (ALARM_SECONDS_REG) Field Descriptions

Bit	Field	Value	Description
31-7	Reserved	0	Reserved
6-4	ALARMSEC1	0-5h	2nd digit of seconds, Range is 0 to 5
3-0	ALARMSEC0	0-9h	1st digit of seconds, Range is 0 to 9

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.3.5.9 Alarm Minute Register (ALARM_MINUTES_REG)

The ALARM_MINUTES_REG is used to program the minute value for the alarm interrupt. Minutes are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.

Figure 19-65. Alarm Minute Register (ALARM_MINUTES_REG)

31–16		
Reserved		
R-0		
15–7	**6–4**	**3–0**
Reserved	ALARM_MIN1	ALARM_MIN0
R-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-70. Alarm Minute Register (ALARM_MINUTES_REG) Field Descriptions

Bit	Field	Value	Description
31-7	Reserved	0	Reserved
6-4	ALARM_MIN1	0-5h	2nd digit of minutes, Range is 0 to 5
3-0	ALARM_MIN0	0-9h	1st digit of minutes, Range is 0 to 9

19.3.5.10 Alarm Hour Register (ALARM_HOURS_REG)

The ALARM_HOURS_REG is used to program the hour value for the alarm interrupt. Hours are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.

Figure 19-66. Alarm Hour Register (ALARM_HOURS_REG)

31–16				
Reserved				
R-0				
15–8	**7**	**6**	**5–4**	**3–0**
Reserved	ALARM_PM_nAM	Rsvd	ALARM_HOUR1	ALARM_HOUR0
R-0	R/W-0	R-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-71. Alarm Hour Register (ALARM_HOURS_REG) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved.
7	ALARM_PM_nAM		Only used in PM_AM mode (otherwise 0)
		0	AM
		1	PM
6	Reserved	0	Reserved.
5-4	ALARM_HOUR1	0-2h	2nd digit of hours, Range is 0 to 2
3-0	ALARM_HOUR0	0-9h	1st digit of hours, Range is 0 to 9

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.3.5.11 Alarm Day of the Month Register (ALARM_DAYS_REG)

The ALARM_DAYS_REG is used to program the day of the month value for the alarm interrupt. Days are stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.

Figure 19-67. Alarm Day of the Month (ALARM_DAYS_REG)

31–16	15–6	5–4	3–0
Reserved	Reserved	ALARM_DAY1	ALARM_DAY0
R-0	R-0	R/W-0	R/W-1

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-72. Alarm Day of the Month Register (ALARM_DAYS_REG) Field Descriptions

Bit	Field	Value	Description
31-6	Reserved	0	Reserved
5-4	ALARM_DAY1	0-3h	2nd digit for days, Range from 0 to 3
3-0	ALARM_DAY0	0-9h	1st digit for days, Range from 0 to 9

19.3.5.12 Alarm Month Register (ALARM_MONTHS_REG)

The ALARM_MONTHS_REG is used to program the month in the year value for the alarm interrupt. The month is stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.

Figure 19-68. Alarm Month Register (ALARM_MONTHS_REG)

31–16	15–5	4	3–0
Reserved	Reserved	ALARM_MONTH1	ALARM_MONTH0
R-0	R-0	R/W-0	R/W-1

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-73. Alarm Month Register (ALARM_MONTHS_REG) Field Descriptions

Bit	Field	Value	Description
31-5	Reserved	0	Reserved
4	ALARM_MONTH1	0-1h	2nd digit of months, Range from 0 to 1
3-0	ALARM_MONTH0	0-9h	1st digit of months, Range from 0 to 9

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.3.5.13 Alarm Year Register (ALARM_YEARS_REG)

The ALARM_YEARS_REG is used to program the year for the alarm interrupt. Only the last two digits are used to represent the year and is stored as BCD format. In BCD format, the decimal numbers 0 through 9 are encoded with their binary equivalent.

Figure 19-69. Alarm Year Register (ALARM_YEARS_REG)

31–16	15–8	7–4	3–0
Reserved	Reserved	ALARM_YEAR1	ALARM_YEAR0
R-0	R-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-74. Alarm Year Register (ALARM_YEARS_REG) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-4	ALARM_YEAR1	0-9h	2nd digit of years, Range from 0 to 9
3-0	ALARM_YEAR0	0-9h	1st digit of years, Range from 0 to 9

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.3.5.14 Control Register (CTRL_REG)

The CTRL_REG contains the controls to enable/disable RTC, set the 12/24 hour time mode, to enable the 30 second rounding feature, and to STOP/START RTC.

Figure 19-70. Control Register (CTRL_REG)

31							8
Reserved							
R-0							
7	6	5	4	3	2	1	0
Rsvd	RTC_disable	SET_32_ COUNTER	TEST_MODE	MODE_12_24	AUTO_COMP	ROUND_30S	STOP_RTC
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-75. Control Register (CTRL_REG) Field Descriptions

Bit	Field	Value	Description
31-7	Reserved	0	Reserved.
6	RTC_disable		Disable RTC module and gate 32-kHz reference clock.
		0	RTC enable
		1	RTC disable (no 32 kHz clock)
5	SET_32_COUNTER		Set the 32-kHz counter with the value stored in the compensation registers when the SET_32_COUNTER bit is set.
		0	No action.
		1	Set the 32Khz counter with compensation registers value
4	TEST_MODE		Test mode.
		0	Functional mode
		1	Test mode (Auto compensation is enabled when the 32Khz counter reaches its end)
3	MODE_12_24		Enable 12-hour mode for HOURS and ALARMHOURS registers.
		0	24-hr mode
		1	12-hour mode
2	AUTO_COMP		Enable oscillator compensation mode.
		0	No auto compensation
		1	Auto compensation enabled
1	ROUND_30S		Enable one-time rounding to nearest minute on next time register read.
		0	No update
		1	Time is rounded to the nearest minute
0	STOP_RTC		Stop the RTC 32-kHz counter.
		0	RTC is frozen
		1	RTC is running

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

NOTE:

- The SET_32_counter must only be used when the RTC is frozen.
- The RTC_DISABLE bit of the CTRL register must only be used to completely disable the RTC function. When this bit is set, the 32 kHz clock is gated, and the RTC is frozen. From this point, resetting this bit to zero can lead to unexpected behavior. This bit should only be used if the RTC function is unwanted in the application, in order to save power.
- MODE_12_24: It is possible to switch between the two modes at any time without disturbing the RTC; read or write is always performed with the current mode.
- Auto compensation is enabled by AUTO_COMP_EN bit in the RTC_CTRL register. If the COMP_REG value is positive, compensation occurs after the second change event. COMP_REG cycles are removed from the next second. If the COMP_REG value is negative, compensation occurs before the second change event. COMP_REG cycles are added to the current second. This enables it to compensate with one 32-kHz period accuracy each hour.
- ROUND_30S is a toggle bit; ARM can only write 1 and RTC clears it. If the ARM sets the ROUND_30S bit and then reads it, the ARM read 1 until the round-to-the-closet-minute is performed at the next second.
- ARM can stop the RTC by clearing STOP_RTC bit of the control register (owing to internal resynchronization, the RUN bit of the status must be checked to be ensure that the RTC is frozen), then update TC values, and re-start the RTC by resetting STOP_RTC bit.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.3.5.15 Status Register (STATUS_REG)

The RTC STATUS_REG contains bits that signal the status of interrupts, events to the processor. Status for the alarm interrupt and timer events are notified by the register.

Figure 19-71. Status Register (STATUS_REG)

31							16
Reserved							
R-0							

15	7	6	5	4	3	2	1	0
Reserved		ALARM	1D_EVENT	1H_EVENT	1M_EVENT	1S_EVENT	RUN	BUSY
R-0		R/W-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-76. Status Register (STATUS_REG) Field Descriptions

Bit	Field	Value	Description
31-7	Reserved	0	Reserved
6	ALARM	0-1	Indicates that an alarm interrupt has been generated
5	1D_EVENT	0-1	One day has occurred
4	1H_EVENT	0-1	One hour has occurred
3	1M_EVENT	0-1	One minute has occurred
2	1S_EVENT	0-1	One second has occurred
1	RUN		RTC is frozen or is running.
		0	RTC is frozen
		1	RTC is running
0	BUSY		Status of RTC module.
		0	Updating event in more than 15 µs
		1	Updating event

NOTE:

- The alarm interrupt keeps its low level until the ARM writes 1 in the ALARM bit of the RTC_STATUS_REG register.
- ALARM: This bit will indicate the status of the alarm interrupt. Writing a 1 to the bit clears the interrupt.
- 1D_EVENT1: This bit will indicate if a day event has occurred. An interrupt will be generated to the processor based on the masking of the interrupt controller.
- 1H_EVENT1: This bit will indicate if an hour event has occurred. An interrupt will be generated to the processor based on the masking of the interrupt controller.
- 1M_EVENT1: This bit will indicate if a minute event has occurred. An interrupt will be generated to the processor based on the masking of the interrupt controller.
- 1S_EVENT1: This bit will indicate if a second event has occurred. An interrupt will be generated to the processor based on the masking of the interrupt controller.
- RUN: This bit will indicate if RTC is frozen or it is running. The RUN bit shows the real state of the RTC. Indeed, because the STOP_RTC signal is resynchronized on 32-kHz clock the action of this bit is delayed.
- BUSY: This bit will give the status of RTC module. The Time and alarm registers can be modified only when this bit is 0.
- The timer interrupt is a negative edge sensitive low-level pulse (1 OCP cycle duration).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.3.5.16 Interrupt Register (INTERRUPTS_REG)

The INTERRUPTS_REG is used to enable or disable RTC from generating interrupts. The timer interrupt and alarm interrupt can be controlled using this register.

Figure 19-72. Interrupt Register (INTERRUPTS_REG)

31				16
Reserved				
R-0				

15	4	3	2	1	0
Reserved		IT_ALARM	IT_TIMER	EVERY	
R-0		R/W-0	R/W-0	R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-77. Interrupt Register (INTERRUPTS_REG) Field Descriptions

Bit	Field	Value	Description
31-4	Reserved	0	Reserved
3	IT_ALARM	0	Enable one interrupt when the alarm value is reached (TC ALARM registers) by the TC registers
2	IT_TIMER		Enable periodic interrupt.
		0	Interrupt disabled
		1	Interrupt enabled
1-0	EVERY	0-3h	Interrupt period.
		0	Every second
		1h	Every minute
		2h	Every hour
		3h	Every day

NOTE:

- The ARM must respect the BUSY period to prevent spurious interrupt. To set a period timer interrupt, the respective period value must be set in the EVERY field. For example, to set a periodic timer interrupt for every hour, the EVERY field has to be set to 2. Along with this the IT_TIMER bit also has to be set for the periodic interrupt to be generated.
- IT_ALARM bit has to be set to generate an alarm interrupt.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.3.5.17 Compensation (LSB) Register (COMP_LSB_REG)

The COMP_LSB_REG is used to program the LSB value of the 32 kHz periods to be added to the 32 kHz counter every hour. This is used to compensate the oscillator drift. The COMP_LSB_REG works together with the compensation (MSB) register (COMP_MSB_REG). The AUTOCOMP bit in the control register (CTRL_REG) must be enabled for compensation to take place.

Figure 19-73. Compensation (LSB) Register (COMP_LSB_REG)

31															16
Reserved															
R-0															

15							8	7							0
Reserved								RTC_COMP_LSB							
R-0								R/W-0							

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-78. Compensation (LSB) Register (COMP_LSB_REG) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-0	RTC_COMP_LSB	0-FFh	Indicates number of 32-kHz periods to be added into the 32-kHz counter every hour

NOTE: This register must be written in two's complement. That means that to add one 32-kHz oscillator period every hour, the ARM must write FFFFh into RTC_COMP_MSB_REG and RTC_COMP_LSB_REG. To remove one 32-kHz oscillator period every hour, the ARM must write 0001h into RTC_COMP_MSB_REG and RTC_COMP_LSB_REG. The 7FFFh value is forbidden.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.3.5.18 Compensation (MSB) Register (COMP_MSB_REG)

The COMP_MSB_REG is used to program the MSB value of the 32 kHz periods to be added to the 32 kHz counter every hour. This is used to compensate the oscillator drift. The COMP_MSB_REG works together with the compensation (LSB) register (COMP_LSB_REG) to set the hourly oscillator compensation value. The AUTOCOMP bit in the control register (CTRL_REG) must be enabled for compensation to take place.

Figure 19-74. Compensation (MSB) Register (COMP_MSB_REG)

31			16
Reserved			
R-0			

15	8	7	0
Reserved		RTC_COMP_MSB	
R-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-79. Compensation (MSB) Register (COMP_MSB_REG) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-0	RTC_COMP_MSB	0-FFh	Indicates number of 32-kHz periods to be added into the 32-kHz counter every hour

NOTE: This register must be written in two's complement. That means that to add one 32-kHz oscillator period every hour, the ARM must write FFFFh into RTC_COMP_MSB_REG and RTC_COMP_LSB_REG. To remove one 32-kHz oscillator period every hour, the ARM must write 0001h into RTC_COMP_MSB_REG and RTC_COMP_LSB_REG. The 7FFFh value is forbidden.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.3.5.19 Oscillator Register (OSC_REG)

The OSC_REG is used to program the oscillator resistance value and the SW_RESET bit can be used to reset the RTC. This is the only way to reset the RTC.

Figure 19-75. Oscillator Register (OSC_REG)

31–16	15–6	5	4	3–0
Reserved	Reserved	SWRESET	OSC32KPWRDNR	SWRESPROG
R-0	R-0	R/W-0	R/W-0(H/W)	R/W-7h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-80. Oscillator Register (OSC_REG) Field Descriptions

Bit	Field	Value	Description
31-6	Reserved	0	Reserved
5	SWRESET	0-1	Software reset bit
4	OSC32KPWRDNR	0-1	Control of 32 kHz Oscillator powerdown
3-0	SWRESPROG	0-Fh	Value of the oscillator resistance

NOTE: SW_RESET bit is set to 1 to reset the RTC module. This bit is self-clearing, and is always read as 0. CPU must care of interrupt handling before RTC is reset. After setting the SW_RESET bit, you must not access any of the RTC registers for 3 32kHz clock cycles (roughly 2200 OCP cycles).

19.3.5.20 Scratch Registers (SCRATCHx_REG)

The SCRATCH_REG is used to hold some required values for the RTC register.

Figure 19-76. Scratch Registers (SCRATCHx_REG)

31–0
RTCSCRATCHx
R/W-0

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 19-81. Scratch Registers (SCRATCHx_REG) Field Descriptions

Bit	Field	Value	Description
31-0	RTCSCRATCHx	0-FFFF FFFFh	Scratch registers, available to program

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.3.5.21 Kick Registers (KICK0R, KICK1R)

The kick registers (KICKnR) are used to enable and disable write protection on the RTC registers. Out of reset, the RTC registers are write-protected. To disable write protection, correct keys must be written to the KICKnR registers.

Kick0 Register (KICK0R)

The Kick0 register allows writing to unlock the kick0 data. To disable RTC register write protection, the value of 83E7 0B13h must be written to KICK0R, followed by the value of 95A4 F1E0h written to KICK1R. RTC register write protection is enabled when any value is written to KICK0R.

Figure 19-77. Kick0 Register (KICK0R)

31	0
KICK0	
W-0	

LEGEND: W = Write only; -*n* = value after reset

Table 19-82. Kick0 Register (KICK0R) Field Descriptions

Bit	Field	Value	Description
31-0	KICK0	0	Kick0 data

Kick1 Register (KICK1R)

The Kick1 register allows writing to unlock the kick1 data and the kicker mechanism to write to other MMRs. To disable RTC register write protection, the value of 83E7 0B13h must be written to KICK0R, followed by the value of 95A4 F1E0h written to KICK1R.

Figure 19-78. Kick1 Register (KICK1R)

31	0
KICK1	
W-0	

LEGEND: W = Write only; -*n* = value after reset

Table 19-83. Kick1 Register (KICK1R) Field Descriptions

Bit	Field	Value	Description
31-0	KICK1	0	Kick1 data

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.3.5.22 RTC Revision Register (RTC_REVISION)

Figure 19-79. RTC Revision Register (RTC_REVISION)

31–30	29–28	27–16
SCHEME	Reserved	FUNC
R-1	R-0	R-EB0h

15–11	10–8	7–6	5–0
R_RTL	X_MAJOR	CUSTOM	Y_MINOR
R-1	R-1	R-0	R-4h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-84. RTC Revision Register (RTC_REVISION) Field Descriptions

Bit	Field	Value	Description
31-30	SCHEME	0-3h	Used to distinguish between old scheme and current
29-28	Reserved	0	Reserved
27-16	FUNC	0-FFFh	Function indicates a software compatible module family
15-11	R_RTL	0-1Fh	RTL Version (R)
10-8	X_MAJOR	0-7h	Major Revision
7-6	CUSTOM	0-3h	Indicates a special version for a particular device
5-0	Y_MINOR	0-3Fh	Minor Revision (Y)

19.3.5.23 System Configuration Register (RTC_SYSCONFIG)

Figure 19-80. System Configuration Register (RTC_SYSCONFIG)

31–16
Reserved
R-0

15–2	1–0
Reserved	IDLEMODE
R-0	R/W-2h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-85. System Configuration Register (RTC_SYSCONFIG) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reserved
1-0	IDLEMODE	0-3h	Configuration of the local target state management mode, By definition target can handle read/write transaction as long as it is out of IDLE state.
		0	Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, i.e., regardless of the IP module's internal requirements; Backup mode, for debug only.
		1h	No-idle mode: local target never enters idle state, Backup mode, for debug only.
		2h	Smart-idle mode: local target's state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements, IP module shall not generate (IRQ- or DMA-request-related) wakeup events.
		3h	Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges the system's idle requests, depending on the IP module's internal requirements, IP module may generate (IRQ- or DMA-request-related) wakeup events when in idle state, Mode is only relevant if the appropriate IP module "swakeup" output(s) is (are) implemented.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.3.5.24 Wakeup Enable Register (RTC_IRQWAKEEN)

Figure 19-81. Wakeup Enable Register (RTC_IRQWAKEEN)

31			16
Reserved			
R-0			

15	2	1	0
Reserved		ALARM_WAKEEN	TIMER_WAKEEN
R-0		R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-86. Wakeup Enable Register (RTC_IRQWAKEEN) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reserved
1	ALARM_WAKEEN		Wakeup generation for event Alarm.
		0	Wakeup disabled
		1	Wakeup enabled
0	TIMER_WAKEEN		Wakeup generation for event Timer.
		0	Wakeup disabled
		1	Wakeup enabled

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.4 WATCHDOG

19.4.1 Introduction

The watchdog timer is an upward counter capable of generating a pulse on the reset pin and an interrupt to the device system modules following an overflow condition. The watchdog timer serves resets to the PRCM module and serves watchdog interrupts to the host ARM and DSP. The reset of the PRCM module causes warm reset of the device.

The watchdog timer can be accessed, loaded, and cleared by registers through the L4 interface. The watchdog timer have the 32-kHz clock for their timer clock input.

The watchdog timer connects to a single target agent port on the L4 interconnect. The default state of the watchdog timer is enabled and not running.

19.4.1.1 Features

The main features of the watchdog timer controllers are:

- L4 slave interface support:
  - 32-bit data bus width
  - 32-/16-bit access supported
  - 8-bit access not supported
  - 11-bit address bus width
  - Burst mode not supported
  - Write nonposted transaction mode only
- Free-running 32-bit upward counter
- Programmable divider clock source (2n where n = 0-7)
- On-the-fly read/write register (while counting)
- Subset programming model of the GP timer
- The watchdog timers are reset either on power-on or after a warm reset before they start counting.
- Reset or interrupt actions when a timer overflow condition occurs
- The watchdog timer generates a reset or an interrupt in its hardware integration.

19.4.1.2 Unsupported Featres

There are no unsupported WD Timer features in this device.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.4.2 Integration

19.4.2.1 6.27.2. Public WD Timer Connectivity Attributes

The general connectivity for the WD Timer module in this device is shown in Table 19-87.

Table 19-87. Public WD Timer Module Connectivity Attributes

Attributes	Type
Power Domain	Wakeup Domain
Clock Domain	PD_WKUP_L4_WKUP_GCLK (OCP) PD_WKUP_WDT1_GCLK (Func)
Reset Signals	WKUP_DOM_RST_N
Idle/Wakeup Signals	Smart Idle / Slave Wakeup
Interrupt Requests	1 Interrupt to Sabertooth and WakeM3
DMA Requests	None
Physical Address	L4 Wakeup slave port

19.4.2.2 6.27.3. Public WD Timer Clock and Reset Management

The Watchdog Timer functional clock (pi_sys_clk input) is sourced from either the on-chip ~32678 Hz oscillator (CLK_RC32K) or the PER PLL generated 32.768 KHz clock (CLK_32KHZ) as selected using CLKSEL_WDT1_CLK[CLKSEL] in the PRCM.

Table 19-88. Public WD Timer Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
PI_OCP_CLK Interface clock	100 MHz	CORE_CLKOUTM4 / 2	pd_wkup_l4_wkup_gclk from PRCM
PI_SYS_CLK Functional clock	32768 Hz	CLK_RC32K or CLK_32KHZ (PER_CLKOUTM2 / 5859.375)	pd_wkup_wdt1_gclk from PRCM

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.4.3 Functional Description

19.4.3.1 Power Management

There are two clock domains in the watchdog timers:

- Functional clock domain: WDTi_FCLK is a 32 kHz watchdog timer functional clock. It is used to clock the watchdog timer internal logic.
- Interface clock domain: WDTi_ICLK is a 125 MHz watchdog timer interface clock. It is used to synchronize the watchdog timer L4 port to the L4 interconnect. All accesses from the interconnect are synchronous to WDTi_ICLK.

In this device, the clocks to the watchdog timers are always On. The clocks cannot be turned off, if the watchdog timers is not being used.

19.4.3.2 Interrupts

Table 19-89 list the event flags, and their masks, that cause module interrupts.

Table 19-89. Watchdog Timer Events

Event Flag	Event Mask	Mapping	Comments
WDT_WIRQSTAT[0] EVENT_OVF	WDT_WIRQENSET/WDT_WIRQENCLR[0] OVF_IT_ENA	WDTINT	Watchdog timer overflow
WDT_WIRQSTAT[1] EVENT_DLY	WDT_WIRQENSET/WDT_WIRQENCLR[1] DLY_IT_ENA	WDTINT	Watchdog delay value reached

19.4.3.3 General Watchdog Timer Operation

The watchdog timers are based on an upward 32-bit counter coupled with a prescaler. The counter overflow is signaled through two independent signals: a simple reset signal and an interrupt signal, both active low. Figure 19-82 is a functional block diagram of the watchdog timer.

The interrupt generation mechanism is controlled through the WDT_WIRQENSET/WDT_WIRQENCLR and WDT_WIRQSTAT registers.

The prescaler ratio can be set from 1 to 128 by accessing the WDT_WCLR[4:2] PTV bit field and the WDT_WCLR[5] PRE bit of the watchdog control register (WDT_WCLR).

The current timer value can be accessed on-the-fly by reading the watchdog timer counter register (WDT_WCRR), modified by accessing the watchdog timer load register (WDT_WLDR) (no on-the-fly update), or reloaded by following a specific reload sequence on the watchdog timer trigger register (WDT_WTGR). A start/stop sequence applied to the watchdog timer start/stop register (WDT_WSPR) can start and stop the watchdog timers.

Figure 19-82. 32-Bit Watchdog Timer Functional Block Diagram

Copyright © 2011, Texas Instruments Incorporated

19.4.3.4 Reset Context

The watchdog timers are enabled after reset. Table 19-90 lists the default reset values of the two watchdog timer load registers (the WDT_WLDR) and prescaler ratios (the WDT_WCLR[4:2] PTV bit field). To get these values, software must read the corresponding WDT_WCLR[4:2] PTV bit field and the 32-bit register to retrieve the static configuration of the module.

Table 19-90. Count and Prescaler Default Reset Values

Timer	WDT_WLDR Reset Value	PTV Reset Value
WDT	FFFF FFBEh	0

19.4.3.5 Overflow/Reset Generation

When the watchdog timer counter register (WDT_WCRR) overflows, an active-low reset pulse is generated to the PRCM module. This RESET pulse causes the PRCM module to generate global WARM reset of the device. It is also driven out of the device through the WD_OUT pin. This pulse is one prescaled timer clock cycle wide and occurs at the same time as the timer counter overflow.

After reset generation, the counter is automatically reloaded with the value stored in the watchdog load register (WDT_WLDR) and the prescaler is reset (the prescaler ratio remains unchanged). When the reset pulse output is generated, the timer counter begins incrementing again.

Figure 19-83 shows a general functional view of the watchdog timers.

Figure 19-83. Watchdog Timers General Functional View

19.4.3.6 Prescaler Value/Timer Reset Frequency

Each watchdog timer is composed of a prescaler stage and a timer counter.

The timer rate is defined by the following values:

- Value of the prescaler fields (the WDT_WCLR[5] PRE bit and the WDT_WCLR[4:2] PTV bit field)
- Value loaded into the timer load register (WDT_WLDR)

The prescaler stage is clocked with the timer clock and acts as a clock divider for the timer counter stage. The ratio is managed by accessing the ratio definition field (the WDT_WCLR[4:2] PTV bit field) and is enabled with the WDT_WCLR[5] PRE bit.

Table 19-91 lists the prescaler clock ratio values.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 19-91. Prescaler Clock Ratio Values

WDT_WCLR[5] PRE	WDT_WCLR[4:2] PTV	Clock Divider (PS)
0	X	1
1	0	1
1	1	2
1	2	4
1	3	8
1	4	16
1	5	32
1	6	64
1	7	128

Thus the watchdog timer overflow rate is expressed as:

OVF_Rate = (FFFF FFFFh – WDT_WLDR + 1) × (wd-functional clock period) × PS

where wd-functional clock period = 1/(wd-functional clock frequency) and PS = $2^{(PTV)}$

CAUTION

Internal resynchronization causes some latency in any software write to WDT_WSPR before WDT_WSPR is updated with the programmed value:

1.5 × functional clock cycles ≤ write_WDT_WSPR_latency ≤ 2.5 × functional clock cycles

Remember to consider this latency whenever the watchdog timer must be started or stopped.

For example, for a timer clock input of 32 kHz with a prescaler ratio value of 1 (clock divided by 2) and WDT_WCLR[5] PRE = 1 (clock divider enabled), the reset period is as listed in Table 19-92.

Table 19-92. Reset Period Examples

WDT_WLDR Value	Reset Period
0000 0000h	74 h 56 min
FFFF 0000h	4 s
FFFF FFF0h	1 ms
FFFF FFFFh	62.5 us

CAUTION

- Ensure that the reloaded value allows the correct operation of the application. When a watchdog timer is enabled, software must periodically trigger a reload before the counter overflows. Hence, the value of the WDT_WLDR[31:0] bit field must be chosen according to the ongoing activity preceding the watchdog reload.
- Due to design reasons, WDT_WLDR[31:0] = FFFF FFFFh is a special case, although such a value of WDT_WLDR is meaningless. When WDT_WLDR is programmed with the overflow value, a triggering event generates a reset/interrupt one functional clock cycle later, even if the watchdog timer is stopped.

Copyright © 2011, Texas Instruments Incorporated

Table 19-93 lists the default reset periods for the watchdog timers.

Table 19-93. Default Watchdog Timer Reset Periods

Watchdog Timers	Clock Source	Default Reset Period
WDT	32 kHz	2 s

19.4.3.7 Triggering a Timer Reload

To reload the timer counter and reset the prescaler before reaching overflow, a reload command is executed by accessing the watchdog timer trigger register (WDT_WTGR) using a specific reload sequence.

The specific reload sequence is performed whenever the written value on the WDT_WTGR register differs from its previous value. In this case, reload is executed in the same way as an overflow autoreload, but without the generation of a reset pulse.

The timer counter is loaded with the value of the watchdog timer load register (the WDT_WLDR[31:0] TIMER_LOAD bit field), and the prescaler is reset.

19.4.3.8 Start/Stop Sequence for Watchdog Timers (Using the WDT_WSPR Register)

To start and stop a watchdog timer, access must be made through the start/stop register (WDT_WSPR) using a specific sequence.

To disable the timer, follow this sequence:

1. Write XXXX AAAAh in WDT_WSPR.
2. Write XXXX 5555h in WDT_WSPR.

To enable the timer, follow this sequence:

1. Write XXXX BBBBh in WDT_WSPR.
2. Write XXXX 4444h in WDT_WSPR.

All other write sequences on the WDT_WSPR register have no effect on the start/stop feature of the module.

19.4.3.9 Modifying Timer Count/Load Values and Prescaler Setting

To modify the timer counter value (the WDT_WCRR register), prescaler ratio (the WDT_WCLR[4:2] PTV bit field), delay configuration value (the WDT_WDLY[31:0] DLY_VALUE bit field), or the load value (the WDT_WLDR[31:0] TIMER_LOAD bit field), the watchdog timer must be disabled by using the start/stop sequence (the WDT_WSPR register).

After a write access, the load register value and prescaler ratio registers are updated immediately, but new values are considered only after the next consecutive counter overflow or after a new trigger command (the WDT_WTGR register).

19.4.3.10 Watchdog Counter Register Access Restriction (WDT_WCRR Register)

A 32-bit shadow register is implemented to read a coherent value of the WDT_WCRR register because the WDT_WCRR register is directly related to the timer counter value and is updated on the timer clock (WDT_FCLK). The shadow register is updated by a 16-bit LSB read command.

NOTE: Although the L4 clock (WDT_ICLK) is completely asynchronous with the timer clock (WDT_FCLK), some synchronization is performed to ensure that the value of the WDT_WCRR register is not read while it is being incremented.

When 32-bit read access is performed, the shadow register is not updated. Read access is performed directly from the accessed register.

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

To ensure that a coherent value is read inside WDT_WCRR, the first read access is to the lower 16 bits (offset = 8h), followed by read access to the upper 16 bits (offset = Ah).

19.4.3.11 Watchdog Timer Interrupt Generation

When an interrupt source occurs, the interrupt status bit (the WDT_WIRQSTAT[0] EVENT_OVF or WDT_WIRQSTAT[1] EVENT_DLY bit) is set to 1. The output interrupt line (WDTi_IRQ) is asserted (active low) when status (the EVENT_xxx bit) and enable (the xxx_IT_ENA bit) flags are set to 1; the order is not relevant. Writing 1 to the enable bit (the status is already set at 1) also triggers the interrupt in the normal order (enable first, status next). The pending interrupt event is cleared when the set status bit is overwritten by a value of 1 by a write command in the WDT_WIRQSTAT register. Reading the WDT_WIRQSTAT register and writing the value back allows a fast interrupt acknowledge process.

The watchdog timer issues an overflow interrupt if this interrupt is enabled in the watchdog interrupt enable register (WDT_WIRQENSET[0] OVF_IT_ENA = 1). When the overflow occurs, the interrupt status bit (the WDT_WIRQSTAT[0] EVENT_OVF bit) is set to 1. The output interrupt line (WDT_IRQ) is asserted (active low) when status (EVENT_OVF) and enable (OVF_IT_ENA) flags are set to 1; the order is not relevant. This interrupt can be disabled by setting the WDT_WIRQENCLR[0] OVF_IT_ENA bit to 1.

The watchdog can issue the delay interrupt if this interrupt is enabled in the interrupt enable register (WDT_WIRQENSET[1] DLY_IT_ENA = 1). When the counter is running and the counter value matches the value stored in the delay configuration register (WDT_WDLY), the corresponding interrupt status bit is set in the watchdog status register (WDT_WIRQSTAT) and the output interrupt line is asserted (active low) when the flag (EVENT_DLY) and enable (DLY_IT_ENA) bits are 1 in the WDT_WIRQSTAT and WDT_WIRQENSET registers, respectively; the order (normally enable, then flag), is not relevant. This interrupt can be disabled by setting the WDT_WIRQENCLR[1] DLY_IT_ENA bit to 1.

NOTE: Writing 0 to the WDT_WIRQSTAT[0] EVENT_OVF bit or the WDT_WIRQSTAT[1] EVENT_DLY bit has no effect.

The two clock domains are resynchronized because the interrupt event is generated on the functional clock domain (WDTi_FCLK) during the updating of the interrupt status register (WDT_WIRQSTAT).

The WDT_WDLY register is used to specify the value of the delay configuration register. The delay time to interrupt is the difference between the reload value stored in the counter load register (WDT_WLDR) and the programmed value in this register (WDT_WDLY).

Use the following formula to estimate the delay time:

Delay time period = (WDT_WDLY – WDT_WLDR + 1) × Timer clock period × Clock divider

Where:

- Timer clock period = 1/(Timer clock frequency)
- Clock divider = 2PTV

If the counter value (WDT_WCRR) reaches the programmed value (WDT_WDLY), the status bit (EVENT_DLY) gets set in the interrupt status register (WDT_WIRQSTAT), and an interrupt occurs if the corresponding enable bit is set in the interrupt enable register (WDT_WIRQENSET).

CAUTION

If the reload event occurs (after a triggering sequence or after a reset sequence) before reaching the programmed value (WDT_WDLY[31:0] WDLY_VALUE), no interrupt is generated.

Also, no interrupt is generated if the value programmed in the delay configuration register (WDT_WDLY) is less than the value stored in the counter load register (WDT_WLDR).

Copyright © 2011, Texas Instruments Incorporated

19.4.3.12 Watchdog Timers Under Emulation

During emulation mode, the watchdog timer can/cannot continue running, according to the value of the WDT_WDSC[5] EMUFREE bit of the system configuration register (WDT_WDSC).

- When EMUFREE is 1, watchdog timer execution is not stopped and a reset pulse is still generated when overflow is reached.
- When EMUFREE is 0, the counters (prescaler/timer) are frozen and incrementation restarts after exiting from emulation mode.

19.4.3.13 Accessing Watchdog Timer Registers

Posted/nonposted selection applies only to functional registers that require synchronization on/from the timer functional clock domain (WDTi_FCLK). For write/read operation, the following registers are affected:

- WDT_WCLR
- WDT_WCRR
- WDT_WLDR
- WDT_WTGR
- WDT_WDLY
- WDT_WSPR

The timer interface clock domain synchronous registers are not affected by the posted/nonposted selection; the write/read operation is effective and acknowledged (command accepted) after one WDT_ICLK cycle from the command assertion. The timer interface clock domain synchronous registers are:

- WDT_WIDR
- WDT_WDSC
- WDT_WDST
- WDT_WIRQSTATRAW
- WDT_WIRQSTAT
- WDT_WIRQENSET
- WDT_WIRQENCLR
- WDT_WWPS

NOTE: Accesses to WDT2 and WDT3 are posted.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.4.3.14 Low-Level Programming Model

This section covers the low-level hardware programming sequences for configuration and use of the module.

19.4.3.14.1 Global Initialization

19.4.3.14.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the watchdog timer is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the watchdog timer (see Table 19-94).

Table 19-94. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	The module interface and functional clocks must be enabled.
Control module	Module-specific pad multiplexing must be set in the control module.
MPU INTC	The MPU INTC configuration must be performed to enable the interrupts from the watchdog timer.

19.4.3.14.1.2 Main Sequence – Watchdog Timer Module Global Initialization

Table 19-95 lists the steps for initializing the watchdog timer module when the module is to be used for the first time.

Table 19-95. Watchdog Timer Module Global Initialization

Step	Register/Bit Field/Programming Model	Value
Execute software reset.	WDT_WDSC[1] SOFTRESET	1
Wait until reset release?	WDT_WDSC[1] SOFTRESET	0
Enable delay interrupt.	WDT_WIRQENSET[1] ENABLE_DLY	1
Enable overflow interrupt.	WDT_WIRQENSET[0] ENABLE_OVF	1

19.4.3.14.2 Operational Mode Configuration

19.4.3.14.2.1 Main Sequence – Watchdog Timer Basic Configuration

Table 19-96 lists the steps for the basic configuration of the watchdog timer.

Table 19-96. Watchdog Timer Basic Configuration

Step	Register/Bit Field/Programming Model	Value
Disable the watchdog timer.	See Section 19.4.3.14.2.2.	
Set prescaler value.	WDT_WCLR[4:2] PTV	xxx
Enable prescaler.	WDT_WCLR[5] PRE	1
Load delay configuration value.	WDT_WDLY	xxx
Load timer counter value.	WDT_WCRR	xxx
Enable the watchdog timer.	See Section 19.4.3.14.2.3.	

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.4.3.14.2.2 Subsequence – Disable the Watchdog Timer

Table 19-97 lists the steps to disable the watchdog timer.

Table 19-97. Disable the Watchdog Timer

Step	Register/Bit Field/Programming Model	Value
Write disable sequence Data1.	WDT_WSPR	XXXX AAAAh
Write disable sequence Data2.	WDT_WSPR	XXXX 5555h

19.4.3.14.2.3 Subsequence – Enable the Watchdog Timer

Table 19-98 lists the steps to enable the watchdog timer.

Table 19-98. Enable the Watchdog Timer

Step	Register/Bit Field/Programming Model	Value
Write enable sequence Data1.	WDT_WSPR	XXXX BBBBh
Write enable sequence Data2.	WDT_WSPR	XXXX 4444h

19.4.4 Use Cases

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.4.5 Watchdog Timer Registers

The watchdog timer registers are listed in Table 19-99.

CAUTION

The watchdog timers registers are limited to 32-bit and 16-bit data accesses; 8-bit access is not allowed and can corrupt register content.

NOTE:

- The WDT_WISR and WDT_WIRQSTATRAW registers have the same functionality. The WDT_WISR register is used for software backward compatibility.
- The WDT_WIER and WDT_WIRQENSET/WDT_WIRQENCLR registers have the same functionality. The WDT_WIER register is used for software backward compatibility.
- The WDT_WIRQSTATRAW and WDT_WIRQSTAT registers give the same information when read. The WDT_WIRQSTATRAW register is used for debug.

Table 19-99. Watchdog Timer Registers

Address Offset	Acronym	Section
0h	WDT_WIDR	Section 19.4.5.1
10h	WDT_WDSC	Section 19.4.5.2
14h	WDT_WDST	Section 19.4.5.3
18h	WDT_WISR	Section 19.4.5.4
1Ch	WDT_WIER	Section 19.4.5.5
20h	WDT_WWER	Section 19.4.5.6
24h	WDT_WCLR	Section 19.4.5.7
28h	WDT_WCRR	Section 19.4.5.8
2Ch	WDT_WLDR	Section 19.4.5.9
30h	WDT_WTGR	Section 19.4.5.10
34h	WDT_WWPS	Section 19.4.5.11
44h	WDT_WDLY	Section 19.4.5.12
48h	WDT_WSPR	Section 19.4.5.13
54h	WDT_WIRQSTATRAW	Section 19.4.5.14
58h	WDT_WIRQSTAT	Section 19.4.5.15
5Ch	WDT_WIRQENSET	Section 19.4.5.16
60h	WDT_WIRQENCLR	Section 19.4.5.17

19.4.5.1 WDT_WIDR Register

The WDT_WIDR register is shown and described in the figure and table below.

Figure 19-84. WDT_WIDR Register

31	0
REVISION	
R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 19-100. WDT_WIDR Register Field Descriptions

Bits	Field	Description
31-0	REVISION	IP Revision

19.4.5.2 WDT_WDSC Register

The WDT_WDSC register is shown and described in the figure and table below. This register controls the various parameters of the L4 interface.

Figure 19-85. WDT_WDSC Register

31 – 6	5	4 – 2	1	0
Reserved	EMUFREE	Reserved	SOFTRESET	Rsvd
R-0	R/W-0	R-0	R/W-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-101. WDT_WDSC Register Field Descriptions

Bits	Field	Description
31-6	Reserved	Write os for future compatibility. Reads return 0.
5	EMUFREE	Emulation mode 0: Timer counter frozen in emulation 1: Timer counter free-running in emulation
4-2	Reserved	Write 0s for future compatibility. Reads return 0.
1	SOFTRESET	Software reset. (Optional) Read 0: Reset done, no pending action Write 0: No action Write 1: Initiate software reset. Read 1: Reset (software or other) ongoing
0	Reserved	Write os for future compatibility. Reads return 0.

19.4.5.3 WDT_WDST Register

The WDT_WDST register provides status information about the module. It is shown and described in the figure and table below.

Figure 19-86. WDT_WDST Register

31 – 1	0
Reserved	RESETDONE
R-0	R-1

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-102. WDT_WDST Register Field Descriptions

Bits	Field	Description
31-1	Reserved	Reads return 0.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 19-102. WDT_WDST Register Field Descriptions (continued)

Bits	Field	Description
0	RESETDONE	Internal module reset monitoring Read 0: Internal module reset is ongoing. Read 1: Reset completed

19.4.5.4 WDT_WISR Register

This register shows which interrupt events are pending inside the module. It is shown and described in the figure and table below.

Figure 19-87. WDT_WISR Register

31	2	1	0
Reserved		DLY_IT_FLAG	OVF_IT_FLAG
R-0		R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-103. WDT_WISR Register Field Descriptions

Bits	Field	Description
31-2	Reserved	Reads return 0.
1	DLY_IT_FLAG	Pending delay interrupt status. Read 0: No delay interrupt pending Write 0: Status unchanged Write 1: Status bit cleared Read 1: Delay interrupt pending
0	OVF_IT_FLAG	Pending overflow interrupt status. Read 0: No overflow interrupt pending Write 0: Status unchanged Write 1: Status bit cleared Read 1: Overflow interrupt pending

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.4.5.5 WDT_WIER Register

The WDT_WIERs register controls (enable/disable) the interrupt events. It is shown and described in the figure and table below.

Figure 19-88. WDT_WIER Register

31	2	1	0
Reserved		DLY_IT_ENA	OVF_IT_ENA
R-0		R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-104. WDT_WIER Register Field Descriptions

Bits	Field	Description
31-2	Reserved	Reads return 0.
1	DLY_IT_ENA	Delay interrupt enable/disable 0: Disable delay interrupt. 1: Enable delay interrupt.
0	OVF_IT_ENA	Overflow interrupt enable/disable 0: Disable overflow interrupt. 1: Enable overflow interrupt.

19.4.5.6 WDT_WWER

TThe WDT_WIERs register controls (enable/disable) the interrupt events. It is shown and described in the figure and table below.

Figure 19-89. WDT_WWER Register

31	2	1	0
Reserved		DLY_WK_ENA	OVF_WK_ENA
R-0		R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-105. WDT_WWER Field Descriptions

Bits	Field	Description
31-2	Reserved	Write 0s for future compatibility. Reads return 0.
1	DLY_WK_ENA	Delay wake-up enable 0: Disable delay wakeup. 1: Enable delay wakeup.
0	OVF_WK_ENA	Overflow wake-up enable 0: Disable overflow wakeup. 1: Enable overflow wakeup.

19.4.5.7 WDT_WCLR Register

The WDT_WCLR register controls the prescaler stage of the counter. It is shown and descirbed in the figure and table below.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 19-90. WDT_WCLR Register

31	6	5	4	2	0
Reserved		PRE	PTV		Rsvd
R-0		R/W-1	R/W-0		R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-106. WDT_WCLR Register Field Descriptions

Bits	Field	Description
31-6	Reserved	Reads return 0.
5	PRE	Prescaler enable/disable configuration 0: Prescaler disabled 1: Prescaler enabled
4-2	PTV	Prescaler value The timer counter is prescaled with the value: 2**PTV. Example: PTV = 3 -> counter increases value if started after 8 functional clock periods. On reset, it is loaded from PI_PTV_RESET_VALUE input port.
1-0	Reserved	Write 0s for future compatibility. Reads return 0.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.4.5.8 WDT_WCRR Register

The WDT_WCRR register holds the value of the internal counter. It is shown and described in the figure and table below.

Figure 19-91. WDT_WCRR Register

31	0
TIMER_COUNTER	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-107. WDT_WCRR Register Field Descriptions

Bits	Field	Description
31-0	TIMER_COUNTER	Value of the timer counter register

19.4.5.9 WDT_WLDR Register

The WDT_WLDR register holds the timer load value. It is shown and described in the figure and table below.

Figure 19-92. WDT_WLDR Register

31	0
TIMER_LOAD	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-108. WDT_WLDR Register Field Descriptions

Bits	Field	Description
31-0	TIMER_LOAD	Value of the timer load register

19.4.5.10 WDT_WTGR Register

Writing a different value than the one already written in this register does a watchdog counter reload.

Figure 19-93. WDT_WTGR Register

31	0
TIGR_VALUE	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-109. WDT_WTGR Register Field Descriptions

Bits	Field	Description
31-0	TTGR_VALUE	Value of the trigger register

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.4.5.11 WDT_WWPS Register

This register contains the write posting bits for all writeable functional registers.

Figure 19-94. WDT_WWPS Register

31							16
Reserved							
R-0							

15	6	5	4	3	2	1	0
Reserved		W_PEND_WDLY	W_PEND_WSPR	W_PEND_WTGR	W_PEND_WLDR	W_PEND_WCRR	W_PEND_WCLR
R-0		R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-110. WDT_WWPS Register Field Descriptions

Bits	Field	Description
31-6	Reserved	Write 0s for future compatibility. Reads return 0.
5	W_PEND_WDLY	Write pending for register WDLY Read 0: No register write pending Read 1: Register write pending
4	W_PEND_WSPR	Write pending for register WSPR Read 0: No register write pending Read 1: Register write pending
3	W_PEND_WTGR	Write pending for register WTGR Read 0: No register write pending Read 1: Register write pending
2	W_PEND_WLDR	Write pending for register WLDR Read 0: No register write pending Read 1: Register write pending
1	W_PEND_WCRR	Write pending for register WCRR Read 0: No register write pending Read 1: Register write pending
0	W_PEND_WCLR	Write pending for register WCLR Read 0: No register write pending Read 1: Register write pending

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.4.5.12 WDT_WDLY Register

ThE WDT_WDLY register holds the delay value that controls the internal pre-overflow event detection.

Figure 19-95. WDT_WDLY

31	0
WDLY_VALUE	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-111. WDT_WDLY Register Field Descriptions

Bits	Field	Description
31-0	WDLY_VALUE	Value of the delay register

19.4.5.13 WDT_WSPR Register

The WDT_WSPR register holds the start-stop value that controls the internal start-stop FSM.

Figure 19-96. WDT_WSPR Register

31	0
WSPR_VALUE	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-112. WDT_WSPR Register Field Descriptions

Bits	Field	Description
31-0	WSPR_VALUE	Value of the start-stop register

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.4.5.14 WDT_WIRQSTATRAW Register

IRQ unmasked status, status set per-event raw interrupt status vector, line 0. Raw status is set even if event is not enabled. Write 1 to set the (raw) status, mostly for debug.

Figure 19-97. WDT_WIRQSTATRAW Register

31	2	1	0
Reserved		EVENT_DLY	EVENT_OVF
R-0		R/W1S-0	R/W1S-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-113. WDT_WIRQSTATRAW Register Field Descriptions

Bits	Field	Description
31-2	Reserved	Write 0s for future compatibility. Reads return 0.
1	EVENT_DLY	Settable raw status for delay event Read 0: No event pending Write 0: No action Write 1: Set event (debug) Read 1: Event pending
0	EVENT_OVF	Settable raw status for overflow event Read 0: No event pending Write 0: No action Write 1: Set event (debug) Read 1: Event pending

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.4.5.15 WDT_WIRQSTAT Register

IRQ masked status, status clear per-event enabled interrupt status vector, line 0. Enabled status is not set unless event is enabled. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled).

Figure 19-98. WIRQSTAT Register

31	2	1	0
Reserved		EVENT_DLY	EVENT_OVF
R-0		R/W1C-0	R/W1C-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-114. WDT_WIRQSTAT RegisterField Descriptions

Bits	Field	Description
31-2	Reserved	Write 0s for future compatibility. Reads return 0.
1	EVENT_DLY	Clearable, enabled status for delay event Read 0: No (enabled) event pending Write 0: No action Write 1: Clear (raw) event Read 1: Event pending
0	EVENT_OVF	Clearable, enabled status for overflow event Read 0: No (enabled) event pending Write 0: No action Write 1: Clear (raw) event Read 1: Event pending

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.4.5.16 WDT_WIRQENSET Register

IRQ enable set per-event interrupt enable bit vector, line 0. Write 1 to set (enable interrupt). Readout equal to corresponding _CLR register.

Figure 19-99. WDT_WIRQENSET Register

31	2	1	0
Reserved		ENABLE_DLY	ENABLE_OVF
R-0		R/W1S-0	R/W1S-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-115. WDT_WIRQENSET Register Field Descriptions

Bits	Field	Description
31-2	Reserved	Write 0s for future compatibility. Reads return 0.
1	ENABLE_DLY	Enable for delay event Read 0: Interrupt disabled (masked) Write 0: No action Write 1: Enable interrupt. Read 1: Interrupt enabled
0	ENABLE_OVF	Enable for overflow event Read 0: Interrupt disabled (masked) Write 0: No action Write 1: Enable interrupt. Read 1: Interrupt enabled

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.4.5.17 WDT_WIRQENCLR Register

IRQ enable clear per-event interrupt enable bit vector, line 0. Write 1 to clear (disable interrupt). Readout equal to corresponding _SET register.

Figure 19-100. WDT_WIRQENCLR Register

31	2	1	0
Reserved		ENABLE_DLY	ENABLE_OVF
R-0		R/W1C-0	R/W1C-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-116. WDT_WIRQENCLR Register Field Descriptions

Bits	Field	Description
31-2	Reserved	Write 0s for future compatibility. Reads return 0.
1	ENABLE_DLY	Enable for delay event Read 0: Interrupt disabled (masked) Write 0: No action Write 1: Disable interrupt. Read 1: Interrupt enabled
0	ENABLE_OVF	Enable for overflow event Read 0: Interrupt disabled (masked) Write 0: No action Write 1: Disable interrupt. Read 1: Interrupt enabled

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

19.4.5.18 WDT_WIRQWAKEEN Register

This register controls (enable/disable) the wake-up events.

Figure 19-101. WIRQWAKEEN Register

31	2	1	0
Reserved		DLY_WK_ENA	OVF_WK_ENA
R-0		R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 19-117. WDT_WIRQWAKEEN Field Descriptions

Bits	Field	Description
31-2	Reserved	Write 0s for future compatibility. Reads return 0.
1	DLY_WK_ENA	Enable delay wake-up 0: Disable delay wakeup 1: Enable delay wakeup
0	OVF_WK_ENA	Enable overflow wakeup 0: Disable overflow wakeup 1: Enable overflow wakeup

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Chapter 20

SPRUH73–July 2011

I2C

This chapter describes the I2C of the device.

Topic **Page**

20.1 Introduction **2610**
20.2 Integration **2611**
20.3 Functional Description **2613**
20.4 I2C Registers **2626**

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.1 Introduction

The multi-master I2C peripheral provides an interface between a CPU and any I2C-bus-compatible device that connects via the I2C serial bus. External components attached to the I2C bus can serially transmit/receive up to 8-bit data to/from the CPU device through the two-wire I2C interface.

The I2C bus is a multi-master bus. The I2C controller does support the multi-master mode that allows more than one device capable of controlling the bus to be connected to it. Each I2C device is recognized by a unique address and can operate as either transmitter or receiver, according to the function of the device. In addition to being a transmitter or receiver, a device connected to the I2C bus can also be considered as master or slave when performing data transfers. Note that a master device is the device which initiates a data transfer on the bus and generates the clock signals to permit that transfer. During this transfer, any device addressed by this master is considered a slave.

20.1.1 I2C Features

The general features of the I2C controller are:

- Compliant with Philips I2C specification version 2.1
- Supports OmniVision Serial Camera Control Bus Protocol (SCCB)
- Supports standard mode (up to 100K bits/s), fast mode (up to 400K bits/s) and high speed mode (up to 3.4Mb/s).
- Multimaster transmitter/slave receiver mode
- Multimaster receiver/slave transmitter mode
- Combined master transmit/receive and receive/transmit modes
- 7-bit and 10-bit device addressing modes
- Built-in FIFO for buffered read or write
  - I2C0: 32 bytes
  - I2C1: 32 bytes
- Programmable clock generation
- Two DMA channels, one interrupt line

20.1.2 Unsupported I2C Features

The I2C module features not supported in this device are shown in Table 20-1.

Table 20-1. Unsupported I2C Features

Feature	Reason
SCCB Protocol	SCCB signal not pinned out
High Speed (3.4 MBPS) operation	Not required

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.2 Integration

This device includes three instantiations of the I2C module. This peripheral implements the multi-master I2C bus which allows serial transfer of 8-bit data to/from other I2C master/slave devices through a two-wire interface. There are three I2C modules instantiations called I2C0, I2C1, and I2C2. The I2C0 module is located in the Wake-up power domain. Figure 20-1 and Figure 20-2 show examples of a system with multiple I2C-compatible devices.

Figure 20-1. I2C0 Integration and Bus Application

Figure 20-2. I2C(1–2) Integration and Bus Application

20.2.1 I2C Connectivity Attributes

The general connectivity attributes for the I2C module are shown in Table 20-2 and Table 20-3.

Table 20-2. I2C0 Connectivity Attributes

Attributes	Type
Power Domain	Wakeup Domain
Clock Domain	PD_WKUP_L4_WKUP_GCLK (OCP) PD_WKUP_I2C0_GFCLK (Func)
Reset Signals	WKUP_DOM_RST_N
Idle/Wakeup Signals	Smart Idle / Wakeup
Interrupt Requests	1 interrupt to MPU Subsystem, PRUSS and WakeM3
DMA Requests	2 DMA requests to EDMA
Physical Address	L4 Wakeup slave port

Table 20-3. I2C(1–2) Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_L4LS_GCLK (OCP) PD_PER_I2C_FCLK (Func)
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	1 interrupt to MPU Subsystem
DMA Requests	2 DMA requests to EDMA
Physical Address	L4 Peripheral slave port

20.2.2 I2C Clock and Reset Management

The I2C controllers have separate bus interface and functional clocks. During power-down mode, the I2C*x*_SCL and I2C*x*_SDA are configured as inputs.

Table 20-4. I2C Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
I2C0 Clock Signals			
PIOCPCLK Interface clock	100 MHz	CORE_CLKOUTM4 / 2	pd_wkup_l4_wkup_gclk From PRCM
PISYSCLK Functional clock	48 MHz	PER_CLKOUTM2 / 4	pd_wkup_i2c0_gfclk From PRCM
I2C(1-2) Clock Signals			
PIOCPCLK Interface clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_l4ls_gclk From PRCM

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Table 20-4. I2C Clock Signals (continued)

Clock Signal	Max Freq	Reference / Source	Comments
PISYSCLK Functional clock	48 MHz	PER_CLKOUTM2 / 4	pd_per_ic2_fclk From PRCM

20.2.3 I2C Pin List

The external signals (I2C*x*_SDA, I2C*x*_SCL) on the device use standard LVCMOS I/Os and may not meet full compliance with the I2C specifications for Fast-mode devices for slope control and input filtering (spike suppression) to improve the EMC behavior.

Table 20-5. I2C Pin List

Pin	Type	Description
I2C*x*_SCL	I/OD	I2C serial clock (open drain)
I2C*x*_SDA	I/OD	I2C serial data (open drain)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.3 Functional Description

20.3.1 Functional Block Diagram

Figure 20-3 shows an example of a system with multiple I2C compatible devices in which the I2C serial ports are all connected together for a two-way transfer from one device to other devices.

Figure 20-3. I2C Functional Block Diagram

The I2C peripheral consists of the following primary blocks:

- A serial interface: one data pin (I2C_SDA) and one clock pin (I2C_SCL).
- Data registers to temporarily hold receive data and transmit data traveling between the I2C_SDA pin and the CPU or the DMA controller.
- Control and status registers
- A peripheral data bus interface to enable the CPU and the DMA controller to access the I2C peripheral registers.
- A clock synchronizer to synchronize the I2C input clock (from the processor clock generator) and the clock on the I2C_SCL pin, and to synchronize data transfers with masters of different clock speeds.
- A prescaler to divide down the input clock that is driven to the I2C peripheral
- A noise filter on each of the two pins, I2C_SDA and I2C_SCL
- An arbitrator to handle arbitration between the I2C peripheral (when it is a master) and another master
- Interrupt generation logic, so that an interrupt can be sent to the CPU
- DMA event generation logic to send an interrupt to the CPU upon reception and data transmission of data.

20.3.2 I2C Master/Slave Contoller Signals

Data is communicated to devices interfacing with the I2C via the serial data line (SDA) and the serial clock line (SCL). These two wires can carry information between a device and others connected to the I2C bus. Both SDA and SCL are bi-directional pins. They must be connected to a positive supply voltage via a pull-up resistor. When the bus is free, both pins are high. The driver of these two pins has an open drain to perform the required wired-AND function.

An example of multiple I2C modules that are connected for a two-way transfer from one device to other devices is shown in Figure 20-4.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 20-4. Multiple I2C Modules Connected

Table 20-6. Signal Pads

	I2C Mode	
Name	**Default Operating Mode**	**Description**
I2C_SCL	In/ Out	I2C serial CLK line Open-drain output buffer. Requires external pull-up resistor (Rp).
I2C_SDA	In/ Out	I2C serial data line Open-drain output buffer. Requires external pull-up resistor (Rp).

20.3.3 I2C Reset

The I2C module can be reset in the following three ways:

- A system reset (PIRSTNA = 0). A device reset causes the system reset. All registers are reset to power up reset values.
- A software reset by setting the SRST bit in the I2C_SYSC register. This bit has exactly the same action on the module logic as the system bus reset. All registers are reset to power up reset values.
- The I2C_EN bit in the I2C_CON register can be used to hold the I2C module in reset. When the system bus reset is removed (PIRSTNA = 1), I2C_EN = 0 keeps the functional part of I2C module in reset state and all configuration registers can be accessed. I2C_EN = 0 does not reset the registers to power up reset values.

Table 20-7. Reset State of I2C Signals

			I2C Reset
Pin	**I/O/Z** [(1)]	**System Reset**	**(I2C_EN = 0)**
SDA	I/O/Z	High impedance	High impedance
SCL	I/O/Z	High impedance	High impedance

(1) I = Input, O = Outpu, Z = High impedance

20.3.4 Data Validity

The data on the SDA line must be stable during the high period of the clock. The high and low states of the data line can only change when the clock signal on the SCL line is LOW.

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Figure 20-5. Bit Transfer on the I2C Bus

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.3.5 START & STOP Conditions

The I2C module generates START and STOP conditions when it is configured as a master.

- START condition is a high-to-low transition on the SDA line while SCL is high.
- STOP condition is a low-to-high transition on the SDA line while SCL is high.
- The bus is considered to be busy after the START condition (BB = 1) and free after the STOP condition (BB = 0).

Figure 20-6. Start and Stop Condition Events

20.3.6 I2C Operation

20.3.6.1 Serial Data Formats

The I2C controller operates in 8-bit word data format (byte write access supported for the last access). Each byte put on the SDA line is 8 bits long. The number of bytes that can be transmitted or received is restricted by the value programmed in the DCOUNT register. The data is transferred with the most significant bit (MSB) first. Each byte is followed by an acknowledge bit from the I2C module if it is in receiver mode.

Figure 20-7. I2C Data Transfer

The I2C module supports two data formats, as shown in Figure 20-8:

- 7-bit/10-bit addressing format
- 7-bit/10-bit addressing format with repeated start condition

The first byte after a start condition (S) always consists of 8 bits. In the acknowledge mode, an extra bit dedicated for acknowledgment is inserted after each byte. In the addressing formats with 7-bit addresses, the first byte is composed of 7 MSB slave address bits and 1 LSB R/nW bit. In the addressing formats with 10-bit addresses, the first byte is composed of 7 MSB slave address bits, such as 11110XX, where XX is the two MSB of the 10-bit addresses, and 1 LSB R/nW bit, which is 0 in this case.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

The least significant R/nW of the address byte indicates the direction of transmission of the following data bytes. If R/nW is 0, the master writes data into the selected slave; if it is 1, the master reads data out of the slave.

Figure 20-8. I2C Data Transfer Formats

20.3.6.2 Master Transmitter

In this mode, data assembled in one of the previously described data formats is shifted out on the serial data line SDA in synch with the self-generated clock pulses on the serial clock line SCL. The clock pulses are inhibited and SCL held low when the intervention of the processor is required (XUDF) after a byte has been transmitted.

20.3.6.3 Master Receiver

This mode can only be entered from the master transmitter mode. With either of the address formats (Figure 6 (a), (b), and (c)), the master receiver is entered after the slave address byte and bit R/W_ has been transmitted, if R/W_ is high. Serial data bits received on bus line SDA are shifted in synch with the self-generated clock pulses on SCL. The clock pulses are inhibited and SCL held low when the intervention of the processor is required (ROVR) after a byte has been transmitted. At the end of a transfer, it generates the stop condition.

20.3.6.4 Slave Transmitter

This mode can only be entered from the slave receiver mode. With either of the address formats (Figure 6 (a), (b), and (c)), the slave transmitter is entered if the slave address byte is the same as its own address and bit R/W_ has been transmitted, if R/W_ is high. The slave transmitter shifts the serial data out on the data line SDA in synch with the clock pulses that are generated by the master device. It does not generate the clock but it can hold clock line SCL low while intervention of the CPU is required (XUDF).

20.3.6.5 Slave Receiver

In this mode, serial data bits received on the bus line SDA are shifted-in in synch with the clock pulses on SCL that are generated by the master device. It does not generate the clock but it can hold clock line SCL low while intervention of the CPU is required (ROVR) following the reception of a byte.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.3.7 Arbitration

If two or more master transmitters start a transmission on the same bus almost simultaneously, an arbitration procedure is invoked. The arbitration procedure uses the data presented on the serial bus by the competing transmitters. When a transmitter senses that a high signal it has presented on the bus has been overruled by a low signal, it switches to the slave receiver mode, sets the arbitration lost (AL) flag, and generates the arbitration lost interrupt. Figure 20-9 shows the arbitration procedure between two devices. The arbitration procedure gives priority to the device that transmits the serial data stream with the lowest binary value. Should two or more devices send identical first bytes, arbitration continues on the subsequent bytes.

Figure 20-9. Arbitration Procedure Between Two Master Transmitters

20.3.8 I2C Clock Generation and I2C Clock Synchronization

Under normal conditions, only one master device generates the clock signal, SCL. During the arbitration procedure, however, there are two or more master devices and the clock must be synchronized so that the data output can be compared. The wired-AND property of the clock line means that a device that first generates a low period of the clock line overrules the other devices. At this high/low transition, the clock generators of the other devices are forced to start generation of their own low period. The clock line is then held low by the device with the longest low period, while the other devices that finish their low periods must wait for the clock line to be released before starting their high periods. A synchronized signal on the clock line is thus obtained, where the slowest device determines the length of the low period and the fastest the length of the high period.

If a device pulls down the clock line for a longer time, the result is that all clock generators must enter the WAIT-state. In this way a slave can slow down a fast master and the slow device can create enough time to store a received byte or to prepare a byte to be transmitted. Figure 20-10 illustrates the clock synchronization.

Figure 20-10. Synchronization of Two I2C Clock Generators

Copyright © 2011, Texas Instruments Incorporated

20.3.9 Prescaler (SCLK/ICLK)

The I2C module is operated with a functional clock (SCLK) frequency that can be in a range of 12-100 MHz, according to I2C mode that must be used (an internal ~24 MHz clock (ICLK) is recommended in case of F/S operation mode e). Note that the frequency of the functional clock influences directly the I2C bus performance and timings.

The internal clock used for I2C logic - ICLK - is generated via the I2C prescaler block. The prescaler consists of a 4-bit register - I2C _PSC, and is used to divide the system clock (SCLK) to obtain the internal required clock for the I2C module.

20.3.10 Noise Filter

The noise filter is used to suppress any noise that is 50 ns or less, in the case of F/S mode of operation. It is designed to suppress noise with one ICLK. The noise filter is always one ICLK cycle, regardless of the bus speed. For FS mode (prescaler = 4, ICLK = 24 MHz), the maximum width of the suppressed spikes is 41.6 ns. To ensure a correct filtering, the prescaler must be programmed accordingly.

20.3.11 I2C Interrupts

The I2C module generates 12 types of interrupt: addressed as slave, bus free (stop condition detected), access error, start condition, arbitration-lost, noacknowledge, general call, registers-ready-for-access, receive and transmit data, receive and transmit draining. These 12 interrupts are accompanied with 12 interrupt masks and flags defined in the I2C_IRQENABLE_SET and respectively I2C_IRQSTATUS_RAW registers. Note that all these 12 interrupt events are sharing the same hardware interrupt line.

- Addressed As Slave interrupt (AAS) is generated to inform the Local Host that an external master addressed the module as a slave. When this interrupt occurs, the CPU can check the I2C_ACTOA status register to check which of the 4 own addresses was used by the external master to access the module.
- Bus Free interrupt (BF) is generated to inform the Local Host that the I2C bus became free (when a Stop Condition is detected on the bus) and the module can initiate his own I2C transaction.
- Start Condition interrupt (STC) is generated after the module being in idle mode have detected (synchronously or asynchronously) a possible Start Condition on the bus (signalized with WakeUp).
- Access Error interrupt (AERR) is generated if a Data read access is performed while RX FIFO is empty or a Data write access is performed while TX FIFO is full.
- Arbitration lost interrupt (AL) is generated when the I2C arbitration procedure is lost.
- No-acknowledge interrupt (NACK) is generated when the master I2C does not receive acknowledge from the receiver.
- General call interrupt (GC) is generated when the device detects the address of all zeros (8 bits).
- Registers-ready-for-access interrupt (ARDY) is generated by the I2C when the previously programmed address, data, and command have been performed and the status bits have been updated. This interrupt is used to let the CPU know that the I2C registers are ready for access.
- Receive interrupt/status (RRDY) is generated when there is received data ready to be read by the CPU from the I2C_DATA register (see the FIFO Management subsection for a complete description of required conditions for interrupt generation). The CPU can alternatively poll this bit to read the received data from the I2C_DATA register.
- Transmit interrupt/status (XRDY) is generated when the CPU needs to put more data in the I2C_DATA register after the transmitted data has been shifted out on the SDA pin (see the FIFO Management subsection for a complete description of required conditions for interrupt generation). The CPU can alternatively poll this bit to write the next transmitted data into the I2C_DATA register.
- Receive draining interrupt (RDR) is generated when the transfer length is not a multiple of threshold value, to inform the CPU that it can read the amount of data left to be transferred and to enable the draining mechanism. (see the Draining Feature subsection for additional details).
- Transmit draining interrupt (XDR) is generated when the transfer length is not a multiple of threshold value, to inform the CPU that it can read the amount of data left to be written and to enable the draining mechanism. (see the Draining Feature subsection for additional details).

Copyright © 2011, Texas Instruments Incorporated

When the interrupt signal is activated, the Local Host must read the I2C_IRQSTATUS_RAW register to define the type of the interrupt, process the request, and then write into these registers the correct value to clear the interrupt flag.

20.3.12 DMA Events

The I2C module can generate two DMA requests events, read (I2C_DMA_RX) and write (I2C_DMA_TX) that can be used by the DMA controller to synchronously read received data from the I2C_DATA or write transmitted data to the I2C_DATA register. The DMA read and write requests are generated in a similar manner as RRDY and XRDY, respectively.

The I2C DMA request signals (I2C_DMA_TX and I2C_DMA_RX) are activated according to the FIFO Management subsection.

20.3.13 Interrupt and DMA Events

I2C has two DMA channels (Tx and Rx). For event numbers, see the data manual.

I2C has one interrupt line for all the interrupt requests. For the interrupt number, see the data manual.

20.3.14 FIFO Management

The I2C module implements two internal 32-bytes FIFOs with dual clock for RX and TX modes. The depth of the FIFOs can be configured at integration via a generic parameter which will also be reflected in I2C_IRQSTATUS_RAW.FIFODEPTH register.

20.3.14.1 FIFO Interrupt Mode Operation

In FIFO interrupt mode (relevant interrupts enabled via I2C_IRQENABLE_SET register), the processor is informed of the status of the receiver and transmitter by an interrupt signal. These interrupts are raised when receive/transmit FIFO threshold (defined by I2C_BUF.TXTRSH or I2C_BUF.RXTRSH) are reached; the interrupt signals instruct the Local Host to transfer data to the destination (from the I2C module in receive mode and/or from any source to the I2C FIFO in transmit mode).

Figure 20-11 and Figure 20-12, respectively, illustrate receive and transmit operations from FIFO management point of view.

Figure 20-11. Receive FIFO Interrupt Request Generation

Note that in Figure 20-11, the RRDY Condition illustrates that the condition for generating a RRDY interrupt is achieved. The interrupt request is generated when this signal is active, and it can be cleared only by the CPU by writing a 1 in the corresponding interrupt flag. If the condition is still present after clearing the previous interrupt, another interrupt request will be generated.

Copyright © 2011, Texas Instruments Incorporated

In receive mode, RRDY interrupt is not generated until the FIFO reaches its receive threshold. Once low, the interrupt can only be de-asserted when the Local Host has handled enough bytes to make the FIFO level below threshold. For each interrupt, the Local Host can be configured to read an amount of bytes equal with the value of the RX FIFO threshold + 1.

Figure 20-12. Transmit FIFO Interrupt Request Generation

Note that in the figure above, the XRDY Condition illustrates that the condition for generating a XRDY interrupt is achieved. The interrupt request is generated when this condition is achieved (when TX FIFO is empty, or the TX FIFO threshold is not reached and there are still data bytes to be transferred in the TX FIFO), and it can be cleared only by the CPU by writing a 1 in the corresponding interrupt flag after transmitting the configured number of bytes. If the condition is still present after clearing the previous interrupt, another interrupt request will be generated.

Note that in interrupt mode, the module offers two options for the CPU application to handle the interrupts:

- When detecting an interrupt request (XRDY or RRDY type), the CPU can write/read one data byte to/from the FIFO and then clear the interrupt. The module will not reassert the interrupt until the interrupt condition is not met.
- When detecting an interrupt request (XRDY or RRDY type), the CPU can be programmed to write/read the amount of data bytes specified by the corresponding FIFO threshold (I2C_BUF.TXTRSH + 1 or I2C_BUF.RXTRSH + 1). In this case, the interrupt condition will be cleared and the next interrupt will be asserted again when the XRDY or RRDY condition will be again met.

If the second interrupt serving approach is used, an additional mechanism (draining feature) is implemented for the case when the transfer length is not a multiple of FIFO threshold (see the Draining Feature subsection).

In slave TX mode, the draining feature cannot be used, since the transfer length is not known at the configuration time, and the external master can end the transfer at any point by not acknowledging one data byte.

20.3.14.2 FIFO Polling Mode Operation

In FIFO polled mode (I2C_IRQENABLE_SET.XRDY_IE and I2C_IRQENABLE_SET.RRDY_IE disabled and DMA disabled), the status of the module (receiver or transmitter) can be checked by polling the XRDY and RRDY status registers (I2C_IRQSTATUS_RAW) (RDR and XDR can also be polled if draining feature must be used). The XRDY and RRDY flags are accurately reflecting the interrupt conditions mentioned in Interrupt Mode. This mode is an alternative to the FIFO interrupt mode of operation, where the status of the receiver and transmitter is automatically known by means of interrupts sent to the CPU.

Copyright © 2011, Texas Instruments Incorporated

20.3.14.3 FIFO DMA Mode Operation

In receive mode, a DMA request is generated as soon as the receive FIFO exceeds its threshold level defined in the threshold level register (I2C_BUF.RXTRSH +1). This request should be de-asserted when the number of bytes defined by the threshold level has been read by the DMA, by setting the I2C_DMARXENABLE_CLR.DMARX_ENABLE_CLEAR field.

Figure 20-13. Receive FIFO DMA Request Generation

In transmit mode, a DMA request is automatically asserted when the transmit FIFO is empty. This request should be de-asserted when the number of bytes defined by the number in the threshold register (I2C_BUF.TXTHRS+1) has been written in the FIFO by the DMA, by setting the I2C_DMATXENABLE_CLR. DMATX_ENABLE_CLEAR field. If an insufficient number of characters are written, then the DMA request will remain active. Figure 20-14 and Figure 20-15 illustrate DMA TX transfers with different values for TXTRSH.

Figure 20-14. Transmit FIFO DMA Request Generation (High Threshold)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 20-15. Transmit FIFO DMA Request Generation (Low Threshold)

Note that also in DMA mode it is possible to have a transfer whose length is not multiple of the configured FIFO threshold. In this case, the DMA draining feature is also used for transferring the additional bytes of the transfer (see the Draining Feature subsection for additional details).

According to the desired operation mode, the programmer must set the FIFO thresholds according to the following table (note that only the OCP side thresholds can be programmed; the I2C side thresholds are default equals to 1). Note that the thresholds must be set consistent with the DMA channel length.

In I2C Slave RX Mode, the Local Host can program the RX threshold with the desired value, and use the FIFO draining feature at the end of the I2C transfer to extract from the FIFO the remaining bytes if the threshold is not reached (see the Draining Feature subsection for additional details).

Note that in I2C Slave TX Mode, the TX FIFO threshold should be set to 1 (I2C_BUF.TXTRSH=0, default value), since the length of the transfer may not be known at configuration time. In this way, the interrupt (or accordingly, DMA) requests will be generated for each byte requested by the remote I2C master to be transferred over the I2C bus. This configuration will prevent the I2C core to request additional data from the CPU or from the DMA controller (using IRQ or DMA), data that will eventually not be extracted from the FIFO by the external master (which can use not acknowledge at any time to end the transfer). If the TX threshold is not set to 1, the module will generate an interrupt or assert a DMA only when the external master requests a byte and the FIFO is empty. However, in this case the TX FIFO will require to be cleared at the end of the transfer.

The I2C module offers the possibility to the user to clear the RX or TX FIFO. This is achieved through I2C_BUF.RXFIFO_CLR and I2C_BUF.TXFIFO_CLR registers, which act like software reset for the FIFOs. In DMA mode, these bits will also reset the DMA state machines.

The FIFO clearing feature can be used when the module is configured as a transmitter, the external receiver responds with a NACK in the middle of the transfer, and there is still data in TX FIFO waiting to be transferred.

On the Functional (I2C) domain, the thresholds can always be considered equal to 1. This means that the I2C Core can start transferring data on the I2C bus whenever it has data in the FIFOs (FIFO is not empty).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.3.14.4 Draining Feature

The Draining Feature is implemented by the I2C core for handling the end of the transfers whose length is not a multiple of FIFO threshold value, and offers the possibility to transfer the remaining amount of bytes (since the threshold is not reached).

Note that this feature prevents the CPU or the DMA controller to attempt more FIFO accesses than necessary (for example, to generate at the end of a transfer a DMA RX request having in the FIFO less bytes than the configured DMA transfer length). Otherwise, an Access Error interrupt will be generated (see I2C_IRQSTATUS_RAW.AERR interrupt).

The Draining mechanism will generate an interrupt (I2C_IRQSTATUS_RAW.RDR or I2C_IRQSTATUS_RAW.XDR) at the end of the transfer informing the CPU that it needs to check the amount of data left to be transferred (I2C_BUFSTAT.TXSTAT or RXSTAT) and to enable the Draining Feature of the DMA controller if DMA mode is enabled (by re-configuring the DMA transfer length according to this value), or perform only the required number of data accesses, if DMA mode is disabled.

In receiving mode (master or slave), if the RX FIFO threshold is not reached but the transfer was ended on the I2C bus and there is still data left in the FIFO (less than the threshold), the receive draining interrupt (I2C_IRQSTATUS_RAW.RDR) will be asserted to inform the local host that it can read the amount of data in the FIFO (I2C_BUFSTAT.RXSTAT). The CPU will perform a number of data read accesses equal with RXSTAT value (if interrupt or polling mode) or re-configure the DMA controller with the required value in order to drain the FIFO.

In master transmit mode, if the TX FIFO threshold is not reached but the amount of data remained to be written in the FIFO is less than TXTRSH, the transmit draining interrupt (I2C_IRQSTATUS_RAW.XDR) will be asserted to inform the local host that it can read the amount of data remained to be written in the TX FIFO (I2C_BUFSTAT.TXSTAT). The CPU will need to write the required number of data bytes (specified by TXSTAT value) or re-configure the DMA controller with the required value in order to transfer the last bytes to the FIFO.

Note that in master mode, the CPU can alternatively skip the checking of TXSTAT and RXSTAT values since it can obtain internally this information (by computing DATACOUNT modulo TX/RXTHRSH).

The draining feature is default disabled, and it can be enabled using I2C_IRQENABLE_SET.XDR_IE or I2C_IRQENABLE_SET.RDR_IE registers (default disabled) only for the transfers with length not equal with the threshold value.

20.3.15 How to Program I2C

20.3.15.1 Module Configuration Before Enabling the Module

1. Program the prescaler to obtain an approximately 12-MHz I2C module clock (I2C_PSC = x; this value is to be calculated and is dependent on the System clock frequency).
2. Program the I2C clock to obtain 100 Kbps or 400 Kbps (SCLL = x and SCLH = x; these values are to be calculated and are dependent on the System clock frequency).
3. Configure its own address (I2C_OA = x) - only in case of I2C operating mode (F/S mode).
4. Take the I2C module out of reset (I2C_CON:I2C_EN = 1).

20.3.15.2 Initialization Procedure

1. Configure the I2C mode register (I2C_CON) bits.
2. Enable interrupt masks (I2C_IRQENABLE_SET), if using interrupt for transmit/receive data.
3. Enable the DMA (I2C_BUF and I2C_DMA/RX/TX/ENABLE_SET) and program the DMA controller) - only in case of I2C operating mode (F/S mode), if using DMA for transmit/receive data.

20.3.15.3 Configure Slave Address and DATA Counter Registers

In master mode, configure the slave address (I2C_SA = x) and the number of byte associated with the transfer (I2C_CNT = x).

Copyright © 2011, Texas Instruments Incorporated

20.3.15.4 Initiate a Transfer

Poll the bus busy (BB) bit in the I2C status register (I2C_IRQSTATUS_RAW). If it is cleared to 0 (bus not busy), configure START/STOP (I2C_CON: STT / I2C_CON: STP condition to initiate a transfer) - only in case of I2C operating mode (F/S mode).

20.3.15.5 Receive Data

Poll the receive data ready interrupt flag bit (RRDY) in the I2C status register (I2C_IRQSTATUS_RAW), use the RRDY interrupt (I2C_IRQENABLE_SET.RRDY_IE set) or use the DMA RX (I2C_BUF.RDMA_EN set together with I2C_DMARXENABLE_SET) to read the receive data in the data receive register (I2C_DATA). Use draining feature (I2C_IRQSTATUS_RAW.RDR enabled by I2C_IRQENABLE_SET.RDR_IE)) if the transfer length is not equal with FIFO threshold.

20.3.15.6 Transmit Data

Poll the transmit data ready interrupt flag bit (XRDY) in the I2C status register (I2C_IRQSTATUS_RAW), use the XRDY interrupt (I2C_IRQENABLE_SET.XRDY_IE set) or use the DMA TX (I2C_BUF.XDMA_EN set together with I2C_DMATXENABLE_SET) to write data into the data transmit register (I2C_DATA). Use draining feature (I2C_IRQSTATUS_RAW.XDR enabled by I2C_IRQENABLE_SET.XDR_IE)) if the transfer length is not equal with FIFO threshold.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.4 I2C Registers

NOTE: All bits defined as reserved must be written by software with 0s, for preserving future compatibility. When read, any reserved bit returns 0. Also, note that it is good software practice to use complete mask patterns for setting or testing individually bit fields within a register.

Table 20-8. I2C Registers

Address Offset	Acronym	Register Name	Section
00h	I2C_REVNB_LO	Module Revision Register (LOW BYTES)	Section 20.4.1
04h	I2C_REVNB_HI	Module Revision Register (HIGH BYTES)	Section 20.4.2
10h	I2C_SYSC	System Configuration Register	Section 20.4.3
20h	I2C_EOI	I2C End of Interrupt Register	Section 20.4.4
24h	I2C_IRQSTATUS_RAW	I2C Status Raw Register	Section 20.4.5
28h	I2C_IRQSTATUS	I2C Status Register	Section 20.4.6
2Ch	I2C_IRQENABLE_SET	I2C Interrupt Enable Set Register	Section 20.4.7
30h	I2C_IRQENABLE_CLR	I2C Interrupt Enable Clear Register	Section 20.4.8
34h	I2C_WE	I2C Wakeup Enable Register	Section 20.4.9
38h	I2C_DMARXENABLE_SET	Receive DMA Enable Set Register	Section 20.4.10
3Ch	I2C_DMATXENABLE_SET	Transmit DMA Enable Set Register	Section 20.4.11
40h	I2C_DMARXENABLE_CLR	Receive DMA Enable Clear Register	Section 20.4.12
44h	I2C_DMATXENABLE_CLR	Transmit DMA Enable Clear Register	Section 20.4.13
48h	I2C_DMARXWAKE_EN	Receive DMA Wakeup Register	Section 20.4.14
4Ch	I2C_DMATXWAKE_EN	Transmit DMA Wakeup Register	Section 20.4.15
90h	2C_SYSS	System Status Register	Section 20.4.16
94h	I2C_BUF	Buffer Configuration Register	Section 20.4.17
98h	I2C_CNT	Data Counter Register	Section 20.4.18
9Ch	I2C_DATA	Data Access Register	Section 20.4.19
A4h	I2C_CON	I2C Configuration Register	Section 20.4.20
A8h	I2C_OA	I2C Own Address Register	Section 20.4.21
ACh	I2C_SA	I2C Slave Address Register	Section 20.4.22
B0h	I2C_PSC	I2C Clock Prescaler Register	Section 20.4.23
B4h	I2C_SCLL	I2C SCL Low Time Register	Section 20.4.24
B8h	I2C_SCLH	I2C SCL High Time Register	Section 20.4.25
BCh	I2C_SYSTEST	System Test Register	Section 20.4.26
C0h	I2C_BUFSTAT	I2C Buffer Status Register	Section 20.4.27
C4h	I2C_OA1	I2C Own Address 1 Register	Section 20.4.28
C8h	I2C_OA2	I2C Own Address 2 Register	Section 20.4.29
CCh	I2C_OA3	I2C Own Address 3 Register	Section 20.4.30
D0h	I2C_ACTOA	Active Own Address Register	Section 20.4.31
D4h	I2C_SBLOCK	I2C Clock Blocking Enable Register	Section 20.4.32

20.4.1 Module Revision Register (LOW BYTES) (I2C_REVNB_LO)

This read-only register contains the hard-coded revision number of the module. A write to this register has no effect.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

NOTE:

- I2C controller with interrupt using interrupt vector register (I2C_IV) is revision 1.x.
- I2C controller with interrupt using status register bits (I2C_IRQSTATUS_RAW) is revision 2.x.

Figure 20-16. Module Revision Register (LOW BYTES) (I2C_REVNB_LO)

31–16	15–11	10–8	7–6	5–0
Reserved	RTL	MAJOR	CUSTOM	MINOR
R-0	R-0	R-0	R-0	R-2h

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20-9. Module Revision Register (LOW BYTES) (I2C_REVNB_LO) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15-11	RTL	0-1Fh	RTL version.
10-8	MAJOR	0-7h	Major Revision. This field changes when there is a major feature change. This field does not change due to bug fix, or minor feature change.
7-6	CUSTOM	0-3h	Indicates a special version for a particular device. Consequence of use may avoid use of standard Chip Support Library (CSL) / Drivers. 0 if non-custom.
5-0	MINOR	0-3Fh	Minor Revision This field changes when features are scaled up or down. This field does not change due to bug fix, or major feature change.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.4.2 Module Revision Register (HIGH BYTES) (I2C_REVNB_HI)

A reset has no effect on the value returned.

Figure 20-17. Module Revision Register (HIGH BYTES) (I2C_REVNB_HI)

31															16
Reserved															
R-0															

15		14	13		12	11									0
SCHEME			Reserved			FUNC									
R-1			R-1			R-40h									

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20-10. Module Revision Register (HIGH BYTES) (I2C_REVNB_HI) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15-14	SCHEME	0-3h	Used to distinguish between old Scheme and current. Spare bit to encode future schemes.
13-12	Reserved	0-3h	Reads return 1h
11-0	FUNC	0-FFFh	Function: Indicates a software compatible module family

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.4.3 System Configuration Register (I2C_SYSC)

This register allows controlling various parameters of the peripheral interface.

Figure 20-18. System Configuration Register (I2C_SYSC)

31															16
Reserved															
R-0															

15	10	9	8	7	5	4	3	2	1	0
Reserved		CLKACTIVITY		Reserved		IDLEMODE		ENAWAKEUP	SRST	AUTOIDLE
R-0		R/W-0		R-0		R/W-0		R/W-0	R/W-0	R/W-1

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 20-11. System Configuration Register (I2C_SYSC) Field Descriptions

Bit	Field	Value	Description
31-10	Reserved	0	Reserved
9-8	CLKACTIVITY		Clock Activity selection bits. Those bits (one bit for each clock signal present on the boundary of the module) are set to 1 to disable external clock gating mechanism in Idle Mode.
		0h	Both clocks can be cut off
		1h	Only OCP clock must be kept active; system clock can be cut off
		2h	Only system clock must be kept active; OCP clock can be cut off
		3h	Both clocks must be kept active
			Values after reset are low (for both 2 bits).
			Note: If the System (functional) Clock is cut-off, the module will assert a WakeUp event when it will asynchronously detect a Start Condition on the I2C Bus. Note that in this case the first transfer will not be taken into account by the module (NACK will be detected by the external master).
7-5	Reserved	0	Reads return 0h.
4-3	IDLEMODE		Idle Mode selection bits. These two bits are used to select one of the idle mode operation mechanisms.
		0h	Force Idle mode
		1h	No Idle mode
		2h	Smart Idle mode
		3h	smartidle_wakeup
			Value after reset is 00 (Force Idle).
2	ENAWAKEUP		Enable Wakeup control bit. When this bit is set to 1, the module enables its own wakeup mechanism.
		0	Wakeup mechanism is disabled
		1	Wakeup mechanism is enabled
			Value after reset is low.
1	SRST		SoftReset bit. When this bit is set to 1, entire module is reset as for the hardware reset. This bit is automatically cleared to 0 by the core and it is only reset by the hardware reset. During reads, it always returns 0.
		0	Normal mode
		1	The module is reset
			Value after reset is low.
0	AUTOIDLE		Autoidle bit. When this bit is set to 1, the module activates its own idle mode mechanism. By evaluating its internal state, the module can decide to gate part of his internal clock tree in order to improve the overall power consumption.
		0	Auto Idle mechanism is disabled
		1	Auto Idle mechanism is enabled
			Value after reset is high.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.4.4 I2C End of Interrupt Register (I2C_EOI)

Figure 20-19. I2C End of Interrupt Register (I2C_EOI)

31	1	0
Reserved		LINE_NUMBER
R-0		W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20-12. I2C End of Interrupt Register (I2C_EOI) Field Descriptions

Bit	Field	Description
31-1	Reserved	Reserved
0	LINE_NUMBER	Software End Of Interrupt (EOI) control. Write number of interrupt output.

20.4.5 I2C Status Raw Register (I2C_IRQSTATUS_RAW)

This register provides core status information for interrupt handling, showing all active events (enabled and not enabled). The fields are read-write. Writing a 1 to a bit will set it to 1, that is, trigger the IRQ (mostly for debug). Writing a 0 will have no effect, that is, the register value will not be modified. Only enabled, active events will trigger an actual interrupt request on the IRQ output line.

Figure 20-20. I2C Status Raw Register (I2C_IRQSTATUS_RAW)

31							16
Reserved							
R-0							
15	14	13	12	11	10	9	8
Reserved	XDR	RDR	BB	ROVR	XUDF	AAS	BF
R-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
AERR	STC	GC	XRDY	RRDY	ARDY	NACK	AL
R/W-0	R/W-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20-13. I2C Status Raw Register (I2C_IRQSTATUS_RAW) Field Descriptions

Bit	Field	Value	Description
31-15	Reserved	0	Write 0s for future compatibility. Read returns 0.
14	XDR		Transmit draining IRQ status. I2C Master Transmit mode only. This read/clear only bit is set to 1 when the module is configured as a master transmitter, the TX FIFO level is below the configured threshold (TXTRSH) and the amount of data still to be transferred is less than TXTRSH. When this bit is set to 1 by the core, CPU must read the I2C_BUFSTAT.TXSTAT register in order to check the amount of data that need to be written in the TX FIFO. Then, according to the mode set (DMA or interrupt), the CPU can enable the DMA draining feature of the DMA controller with the number of data bytes to be transferred (I2C_BUFSTAT.TXSTAT), or generate write data accesses according to this value (IRQ mode). The interrupt needs to be cleared after the DMA controller was reconfigured (if DMA mode enabled), or before generating data accesses to the FIFO (if IRQ mode enabled). If the corresponding interrupt was enabled, an interrupt is signaled to the local host. The CPU can also poll this bit. For more details about TDR generation, refer to the FIFO Management subsection. The CPU can only clear this bit by writing a 1 into this register. A write 0 has no effect.
		0	Transmit draining inactive
		1	Transmit draining enabled
			Value after reset is low.

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Table 20-13. I2C Status Raw Register (I2C_IRQSTATUS_RAW) Field Descriptions (continued)

Bit	Field	Value	Description
13	RDR		Receive draining IRQ status. I2C Receive mode only. This read/clear only bit is set to 1 when the module is configured as a receiver, a stop condition was received on the bus and the RX FIFO level is below the configured threshold (RXTRSH). When this bit is set to 1 by the core, CPU must read the I2C_BUFSTAT.RXSTAT register in order to check the amount of data left to be transferred from the FIFO. Then, according to the mode set (DMA or interrupt), the CPU needs to enable the draining feature of the DMA controller with the number of data bytes to be transferred (I2C_BUFSTAT.RXSTAT), or generate read data accesses according to this value (IRQ mode). The interrupt needs to be cleared after the DMA controller was reconfigured (if DMA mode enabled), or before generating data accesses to the FIFO (if IRQ mode enabled). If the corresponding interrupt was enabled, an interrupt is signaled to the local host. The CPU can also poll this bit. For more details about RDR generation, refer to the FIFO Management subsection. The CPU can only clear this bit by writing a 1 into this register. A write 0 has no effect.
		0	Receive draining inactive
		1	Receive draining enabled
			Value after reset is low.
12	BB		This read-only bit indicates the state of the serial bus. In slave mode, on reception of a start condition, the device sets BB to 1. BB is cleared to 0 after reception of a stop condition. In master mode, the software controls BB. To start a transmission with a start condition, MST, TRX, and STT must be set to 1 in the I2C_CON register. To end a transmission with a stop condition, STP must be set to 1 in the I2C_CON register. When BB = 1 and STT = 1, a restart condition is generated.
		0	Bus is free
		1	Bus is occupied
			Value after reset is low.
11	ROVR		Receive overrun status. Writing into this bit has no effect. I2C receive mode only. This read-only bit indicates whether the receiver has experienced overrun. Overrun occurs when the shift register is full and the receive FIFO is full. An overrun condition does not result in a data loss; the peripheral is just holding the bus (low on SCL) and prevents other bytes from being received. ROVR is set to 1 when the I2C has recognized an overrun. ROVR is clear when reading I2C_DATA register, or when resetting the I2C (I2C_CON:I2C_EN = 0).
		0	Normal operation
		1	Receiver overrun
			Value after reset is low.
10	XUDF		Transmit underflow status. Writing into this bit has no effect. I2C transmit mode only. This read-only bit indicates whether the transmitter has experienced underflow. In master transmit mode, underflow occurs when the shift register is empty, the transmit FIFO is empty, and there are still some bytes to transmit (DCOUNT 0). In slave transmit mode, underflow occurs when the shift register is empty, the transmit FIFO is empty, and there are still some bytes to transmit (read request from external I2C master). XUDF is set to 1 when the I2C has recognized an underflow. The core holds the line till the underflow cause has disappeared. XUDF is clear when writing I2C_DATA register or resetting the I2C (I2C_CON:I2C_EN = 0).
		0	Normal operation
		1	Transmit underflow
			Value after reset is low.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 20-13. I2C Status Raw Register (I2C_IRQSTATUS_RAW) Field Descriptions (continued)

Bit	Field	Value	Description
9	AAS		Address recognized as slave IRQ status. I2C mode only. This read only bit is set to 1 by the device when it has recognized its own slave address (or one of the alternative own addresses), or an address of all zeros (8 bits). When this bit is set to 1 by the core, an interrupt is signaled to the local host if the interrupt was enabled. This bit can be cleared in 2 ways: • If the interrupt was enabled, it will be cleared by writing 1 into this register (writing 0 has no effect) • If the interrupt was not enabled, the AAS bit is reset to 0 by restart or stop
		0	No action
		1	Address recognized
			Value after reset is low.
8	BF		I2C mode only. This read only bit is set to 1 by the device when the I2C bus became free (after a transfer is ended on the bus – stop condition detected). This interrupt informs the Local Host that it can initiate its own I2C transfer on the bus. When this bit is set to 1 by the core, an interrupt is signaled to the local host if the interrupt was enabled. The CPU can only clear this bit by writing a 1 into this register. Writing 0 has no effect.
		0	No action
		1	Bus Free
			Value after reset is low.
7	AERR		Access Error IRQ status. I2C mode only. This read/clear only bit is set to 1 by the device if an OCP write access is performed to I2C_DATA while the TX FIFO is full or if an OCP read access is performed to the I2C_DATA while the RX FIFO is empty. Note that, when the RX FIFO is empty, a read access will return to the previous read data value. When the TX FIFO is full, a write access is ignored. In both events, the FIFO pointers will not be updated. When this bit is set to 1 by the core, an interrupt is signaled to the local host if the interrupt was enabled. The CPU can only clear this bit by writing a 1 into this register. Writing 0 has no effect.
		0	No action
		1	Access Error
			Value after reset is low.
6	STC		Start Condition IRQ status. I2C mode only. This read/clear only bit is set to 1 by the device if previously the module was in idle mode and a start condition was asynchronously detected on the I2C Bus and signalized with an Wakeup (if the I2C_SYSC.ClockActivity allows the system clock to be cut-off). When the Active Mode will be restored and the interrupt generated, this bit will indicate the reason of the wakeup. **Note 1:** The corresponding interrupt for this bit should be enabled only if the module was configured to allow the possibility of cutting-off the system clock while in Idle State (I2C_SYSC.ClockActivity = 00 or 01). **Note 2:** The first transfer (corresponding to the detected start condition) will be lost (not taken into account by the module) and it will be used only for generating the WakeUp enable for restoring the Active Mode of the module. On the I2C line, the external master which generated the transfer will detect this behavior as a not acknowledge to the address phase and will possibly restart the transfer. The CPU can only clear this bit by writing a 1 into this register. Writing 0 has no effect.
		0	No action
		1	Start Condition detected
			Value after reset is low.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 20-13. I2C Status Raw Register (I2C_IRQSTATUS_RAW) Field Descriptions (continued)

Bit	Field	Value	Description
5	GC		General call IRQ status. Set to '1' by core when General call address detected and interrupt signaled to MPUSS. Write '1' to clear. I2C mode only. This read/clear only bit is set to 1 by the device if it detects the address of all zeros (8 bits) (general call). When this bit is set to 1 by the core, an interrupt is signaled to the local host if the interrupt was enabled. The CPU can only clear this bit by writing a 1 into this register. Writing 0 has no effect. **Note:** When this bit is set to 1, AAS also reads as 1.
		0	No general call detected
		1	General call address detected
			Value after reset is low.
4	XRDY		Transmit data ready IRQ status. Set to '1' by core when transmitter and when new data is requested. When set to '1' by core, an interrupt is signaled to MPUSS. Write '1' to clear. Transmit mode only (I2C mode). This read/clear only bit (XRDY) is set to 1 when the I2C peripheral is a master or slave transmitter, the CPU needs to send data through the I2C bus, and the module (transmitter) requires new data to be served. Note that a master transmitter requests new data if the FIFO TX level is below the threshold (TXTRSH) and the required amount of data remained to be transmitted (I2C_BUFSTAT.TXSTAT) is greater than the threshold. A slave transmitter requests new data when the FIFO TX level is below the threshold (if TXTRSH > 1), or anytime there is a read request from external master (for each acknowledge received from the master), if TXTRSH = 1. When this bit is set to 1 by the core, an interrupt is signaled to the local host if the interrupt was enabled. The CPU can also poll this bit (refer to the FIFO Management subsection for details about XRDY generation). The CPU can only clear this bit by writing a 1 into this register. Writing 0 has no effect. **Note:** If the DMA transmit mode is enabled (I2C_BUF.XDMA_EN is set, together with I2C_DMATXENABLE_SET), this bit is forced to 0 and no interrupt will be generated; instead, a DMA TX request to the main DMA controller of the system is generated.
		0	Transmission ongoing
		1	Transmit data ready
			Value after reset is low.
3	RRDY		Receive mode only (I2C mode). This read/clear only RRDY is set to 1 when the RX FIFO level is above the configured threshold (RXTRSH). When this bit is set to 1 by the core, CPU is able to read new data from the I2C_DATA register. If the corresponding interrupt was enabled, an interrupt is signaled to the local host. The CPU to read the received data in I2C_DATA register can also poll this bit (refer to the FIFO Management subsection for details about RRDY generation). The CPU can only clear this bit by writing a 1 into this register. A write 0 has no effect. If the DMA receive mode is enabled (I2C_BUF.RDMA_EN is set, together with I2C_DMARXENABLE_SET), this bit is forced to 0 and no interrupt will be generated; instead a DMA RX request to the main DMA controller of the system is generated.
		0	Receive FIFO threshold not reached
		1	Receive data ready for read (RX FIFO threshold reached)
			Value after reset is low.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 20-13. I2C Status Raw Register (I2C_IRQSTATUS_RAW) Field Descriptions (continued)

Bit	Field	Value	Description
2	ARDY		I2C mode only. This read/clear only bit, when set to 1, indicates that the previously programmed data and command (receive or transmit, master or slave) has been performed and status bit has been updated. The CPU uses this flag to let it know that the I2C registers are ready to be accessed again. The CPU can only clear this bit by writing a 1 into this register. A write 0 has no effect. **Mode** / **Others** / **ARDY Set Condition** I2C Master transmit / STP = 1 / DCOUNT = 0 I2C Master receive / STP = 1 / DCOUNT = 0 and receiver FIFO empty I2C Master transmit / STP = 0 / DCOUNT passed 0 I2C Master receive / STP = 0 / DCOUNT passed 0 and receiver FIFO empty I2C Master transmit / - / Stop or restart condition received from master I2C Slave receive / - / Stop or restart condition and receiver FIFO empty
		0	No action
		1	Access ready
			Value after reset is low.
1	NACK		No acknowledgement IRQ status. Bit is set when No Acknowledge has been received, an interrupt is signaled to MPUSS. Write '1' to clear this bit. I2C mode only. The read/clear only No Acknowledge flag bit is set when the hardware detects No Acknowledge has been received. When a NACK event occurs on the bus, this bit is set to 1, the core automatically ends the transfer and clears the MST/STP bits in the I2C_CON register and the I2C becomes a slave. Clearing the FIFOs from remaining data might be required. The CPU can only clear this bit by writing a 1 into this register. Writing 0 has no effect.
		0	Normal operation
		1	Not Acknowledge detected
			Value after reset is low.
0	AL		Arbitration lost IRQ status. This bit is automatically set by the hardware when it loses the Arbitration in master transmit mode, an interrupt is signaled to MPUSS. During reads, it always returns 0. I2C mode only. The read/clear only Arbitration Lost flag bit is set to 1 when the device (configured in master mode) detects it has lost an arbitration (in Address Phase). This happens when two or more masters initiate a transfer on the I2C bus almost simultaneously or when the I2C attempts to start a transfer while BB (bus busy) is 1. When this is set to 1 due to arbitration lost, the core automatically clears the MST/STP bits in the I2C_CON register and the I2C becomes a slave receiver. The CPU can only clear this bit by writing a 1 to this register. Writing 0 has no effect.
		0	Normal operation
		1	Arbitration lost detected
			Value after reset is low.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.4.6 I2C Status Register (I2C_IRQSTATUS)

This register provides core status information for interrupt handling, showing all active and enabled events and masking the others. The fields are read-write. Writing a 1 to a bit will clear it to 0, that is, clear the IRQ. Writing a 0 will have no effect, that is, the register value will not be modified. Only enabled, active events will trigger an actual interrupt request on the IRQ output line.

For all the internal fields of the I2C_IRQSTATUS register, the descriptions given in the I2C_IRQSTATUS_RAW subsection are valid.

Figure 20-21. I2C Status Register (I2C_IRQSTATUS)

31							16
Reserved							
R-0							
15	14	13	12	11	10	9	8
Reserved	XDR	RDR	BB	ROVR	XUDF	AAS	BF
R-0	R/W1C-0	R/W1C-0	R/W-0	R/W-0	R/W1C-0	R/W1C-0	R/W1C-0
7	6	5	4	3	2	1	0
AERR	STC	GC	XRDY	RRDY	ARDY	NACK	AL
R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20-14. I2C Status Register (I2C_IRQSTATUS) Field Descriptions

Bit	Field	Value	Description
31-15	Reserved	0	Write 0s for future compatibility. Read returns 0.
14	XDR		Transmit draining IRQ enabled status.
		0	Transmit draining inactive
		1	Transmit draining enabled
13	RDR		Receive draining IRQ enabled status.
		0	Receive draining inactive
		1	Receive draining enabled
12	BB		Bus busy enabled status.Writing into this bit has no effect.
		0	Bus is free
		1	Bus is occupied
11	ROVR		Receive overrun enabled status.Writing into this bit has no effect.
		0	Normal operation
		1	Receiver overrun
10	XUDF		Transmit underflow enabled status.Writing into this bit has no effect.
		0	Normal operation
		1	Transmit underflow
9	AAS		Address recognized as slave IRQ enabled status.
		0	No action
		1	Address recognized
8	BF		Bus Free IRQ enabled status.
		0	No action
		1	Bus free
7	AERR		Access Error IRQ enabled status.
		0	No action
		1	Access error
6	STC		Start Condition IRQ enabled status.
		0	No action
		1	Start condition detected

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 20-14. I2C Status Register (I2C_IRQSTATUS) Field Descriptions (continued)

Bit	Field	Value	Description
5	GC		General call IRQ enabled status. Set to '1' by core when General call address detected and interrupt signaled to MPUSS. Write '1' to clear.
		0	No general call detected
		1	General call address detected
4	XRDY		Transmit data ready IRQ enabled status. Set to '1' by core when transmitter and when new data is requested. When set to '1' by core, an interrupt is signaled to MPUSS. Write '1' to clear.
		0	Transmision ongoing
		1	Transmit data ready
3	RRDY		Receive data ready IRQ enabled status. Set to '1' by core when receiver mode, a new data is able to be read. When set to '1' by core, an interrupt is signaled to MPUSS. Write '1' to clear.
		0	No data available
		1	Receive data available
2	ARDY		Register access ready IRQ enabled status. When set to '1' it indicates that previous access has been performed and registers are ready to be accessed again. An interrupt is signaled to MPUSS. Write '1' to clear.
		0	Module busy
		1	Access ready
1	NACK		No acknowledgement IRQ enabled status. Bit is set when No Acknowledge has been received, an interrupt is signaled to MPUSS. Write '1' to clear this bit.
		0	Normal operation
		1	Not Acknowledge detected
0	AL		Arbitration lost IRQ enabled status. This bit is automatically set by the hardware when it loses the Arbitration in master transmit mode, an interrupt is signaled to MPUSS. During reads, it always returns 0.
		0	Normal operation
		1	Arbitration lost detected

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.4.7 I2C Interrupt Enable Set Register (I2C_IRQENABLE_SET)

All 1-bit fields enable a specific interrupt event to trigger an interrupt request. Writing a 1 to a bit will enable the field. Writing a 0 will have no effect, that is, the register value will not be modified.

For all the internal fields of the I2C_IRQENABLE_SET register, the descriptions given in the I2C_IRQSTATUS_RAW subsection are valid.

Figure 20-22. I2C Interrupt Enable Set Register (I2C_IRQENABLE_SET)

31							16
Reserved							
R-0							
15	14	13	12	11	10	9	8
Reserved	XDR_IE	RDR_IE	Reserved	ROVR	XUDF	AAS_IE	BF_IE
R-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
AERR_IE	STC_IE	GC_IE	XRDY_IE	RRDY_IE	ARDY_IE	NACK_IE	AL_IE
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20-15. I2C Interrupt Enable Set Register (I2C_IRQENABLE_SET) Field Descriptions

Bit	Field	Value	Description
31-15	Reserved	0	Write 0s for future compatibility. Read returns 0.
14	XDR_IE		Transmit draining interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[XDR].
		0	Transmit draining interrupt disabled
		1	Transmit draining interrupt enabled
13	RDR_IE		Receive draining interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[RDR].
		0	Receive draining interrupt disabled
		1	Receive draining interrupt enabled
12	Reserved		Reserved
11	ROVR		Receive overrun enable set.
		0	Receive overrun interrupt disabled
		1	Receive draining interrupt enabled
10	XUDF		Transmit underflow enable set.
		0	Transmit underflow interrupt disabled
		1	Transmit underflow interrupt enabled
9	AAS_IE		Addressed as slave interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[AAS].
		0	Addressed as slave interrupt disabled
		1	Addressed as slave interrupt enabled
8	BF_IE		Bus free interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[BF].
		0	Bus free interrupt disabled
		1	Bus free interrupt enabled
7	AERR_IE		Access error interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[AERR].
		0	Access error interrupt disabled
		1	Access error interrupt enabled
6	STC_IE		Start condition interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[STC].
		0	Start condition interrupt disabled
		1	Start condition interrupt enabled

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 20-15. I2C Interrupt Enable Set Register (I2C_IRQENABLE_SET) Field Descriptions (continued)

Bit	Field	Value	Description
5	GC_IE		General call interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[GC].
		0	General call interrupt disabled
		1	General call interrupt enabled
4	XRDY_IE		Transmit data ready interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[XRDY].
		0	Transmit data ready interrupt disabled
		1	Transmit data ready interrupt enabled
3	RRDY_IE		Receive data ready interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[RRDY]
		0	Receive data ready interrupt disabled
		1	Receive data ready interrupt enabled
2	ARDY_IE		Register access ready interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[ARDY].
		0	Register access ready interrupt disabled
		1	Register access ready interrupt enabled
1	NACK_IE		No acknowledgement interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[NACK].
		0	Not Acknowledge interrupt disabled
		1	Not Acknowledge interrupt enabled
0	AL_IE		Arbitration lost interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[AL].
		0	Arbitration lost interrupt disabled
		1	Arbitration lost interrupt enabled

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.4.8 I2C Interrupt Enable Clear Register (I2C_IRQENABLE_CLR)

All 1-bit fields clear a specific interrupt event. Writing a 1 to a bit will disable the interrupt field. Writing a 0 will have no effect, that is, the register value will not be modified.

For all the internal fields of the I2C_IRQENABLE_CLR register, the descriptions given in the I2C_IRQSTATUS_RAW subsection are valid.

Figure 20-23. I2C Interrupt Enable Clear Register (I2C_IRQENABLE_CLR)

31							16
Reserved							
R-0							
15	14	13	12	11	10	9	8
Reserved	XDR_IE	RDR_IE	Reserved	ROVR	XUDF	AAS_IE	BF_IE
R-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
AERR_IE	STC_IE	GC_IE	XRDY_IE	RRDY_IE	ARDY_IE	NACK_IE	AL_IE
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20-16. I2C Interrupt Enable Clear Register (I2C_IRQENABLE_CLR) Field Descriptions

Bit	Field	Value	Description
31-15	Reserved	0	Write 0s for future compatibility. Read returns 0.
14	XDR_IE		Transmit draining interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[XDR].
		0	Transmit draining interrupt disabled
		1	Transmit draining interrupt enabled
13	RDR_IE		Receive draining interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[RDR].
		0	Receive draining interrupt disabled
		1	Receive draining interrupt enabled
12	Reserved		Reserved
11	ROVR		Receive overrun enable clear.
		0	Receive overrun interrupt disabled
		1	Receive draining interrupt enabled
10	XUDF		Transmit underflow enable clear.
		0	Transmit underflow interrupt disabled
		1	Transmit underflow interrupt enabled
9	AAS_IE		Addressed as slave interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[AAS].
		0	Addressed as slave interrupt disabled
		1	Addressed as slave interrupt enabled
8	BF_IE		Bus Free interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[BF].
		0	Bus free interrupt disabled
		1	Bus free interrupt enabled
7	AERR_IE		Access error interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[AERR].
		0	Access error interrupt disabled
		1	Access error interrupt enabled
6	STC_IE		Start condition interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[STC].
		0	Start condition interrupt disabled
		1	Start condition interrupt enabled

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 20-16. I2C Interrupt Enable Clear Register (I2C_IRQENABLE_CLR) Field Descriptions (continued)

Bit	Field	Value	Description
5	GC_IE		General call interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[GC].
		0	General call interrupt disabled
		1	General call interrupt enabled
4	XRDY_IE		Transmit data ready interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[XRDY].
		0	Transmit data ready interrupt disabled
		1	Transmit data ready interrupt enabled
3	RRDY_IE		Receive data ready interrupt enable set. Mask or unmask the interrupt signaled by bit in I2C_STAT[RRDY]
		0	Receive data ready interrupt disabled
		1	Receive data ready interrupt enabled
2	ARDY_IE		Register access ready interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[ARDY].
		0	Register access ready interrupt disabled
		1	Register access ready interrupt enabled
1	NACK_IE		No acknowledgement interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[NACK].
		0	Not Acknowledge interrupt disabled
		1	Not Acknowledge interrupt enabled
0	AL_IE		Arbitration lost interrupt enable clear. Mask or unmask the interrupt signaled by bit in I2C_STAT[AL].
		0	Arbitration lost interrupt disabled
		1	Arbitration lost interrupt enabled

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.4.9 I2C Wakeup Enable Register (I2C_WE)

Every 1-bit field in the I2C_WE register enables a specific (synchronous) IRQ request source to generate an asynchronous wakeup (on the appropriate swakeup line). When a bit location is set to 1 by the local host, a wakeup is signaled to the local host if the corresponding event is captured by the core of the I2C controller. Value after reset is low (all bits).

NOTE:

- There is no need for an Access Error WakeUp event, since this event occurs only when the module is in Active Mode (for OCP accesses to FIFO) and is signaled by an interrupt.
- With the exception of Start Condition WakeUp, which is asynchronously detected when the Functional clock is turned-off, all the other WakeUp events require the Functional (System) clock to be enabled.

Figure 20-24. I2C Wakeup Enable Register (I2C_WE)

31							16
Reserved							
R-0							
15	14	13	12	11	10	9	8
Reserved	XDR_WE	RDR_WE	Reserved	ROVR_WE	XUDF_WE	AAS_WE	BF_WE
R-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
Reserved	STC_WE	GC_WE	Reserved	DRDY_WE	ARDY_WE	NACK_WE	AL_WE
R-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20-17. I2C Wakeup Enable Register (I2C_WE) Field Descriptions

Bit	Field	Value	Description
31-15	Reserved	0	Reserved
14	XDR_WE		Transmit draining wakeup enable. This read/write bit is used to enable or disable wakeup signal generation when I2C module is in idle mode, the TX FIFO level is below the threshold and the amount of data left to be transferred is less than TXTRSH value. This allows for the module to inform the CPU that it can check the amount of data to be written to the FIFO.
		0	Transmit draining wakeup disabled
		1	Transmit draining wakeup enabled
13	RDR_WE		Receive draining wakeup enable. This read/write bit is used to enable or disable wakeup signal generation when I2C is in idle mode, configured as a receiver, and it has detected a stop condition on the bus but the RX FIFO threshold is not reached (but the FIFO is not empty). This allows for the module to inform the CPU that it can check the amount of data to be transferred from the FIFO.
		0	Receive draining wakeup disabled
		1	Receive draining wakeup enabled
12	Reserved	0	Reserved
11	ROVR_WE		Receive overrun wakeup enable
		0	Receive overrun wakeup disabled
		1	Receive overrun wakeup enabled
10	XUDF_WE		Transmit underflow wakeup enable
		0	Transmit underflow wakeup disabled
		1	Transmit underflow wakeup enabled

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 20-17. I2C Wakeup Enable Register (I2C_WE) Field Descriptions (continued)

Bit	Field	Value	Description
9	AAS_WE		Address as slave IRQ wakeup enable. This read/write bit is used to enable or disable wakeup signal generation when I2C module is in idle mode, and external master addresses the I2C module as a slave. This allows for the module to inform the CPU that it can check which of the own addresses was used by the external master to access the I2C core.
		0	Addressed as slave wakeup disabled
		1	Addressed as slave wakeup enabled
8	BF_WE		Bus free IRQ wakeup enable. This read/write bit is used to enable or disable wakeup signal generation when I2C module is in idle mode and the I2C bus became free. This allows for the module to inform the CPU that it can initiate its own transfer on the I2C line.
		0	Bus Free wakeup disabled
		1	Bus Free wakeup enabled
7	Reserved	0	Reserved
6	STC_WE		Start condition IRQ wakeup set. This read/write bit is used to enable or disable wakeup signal generation when I2C module is in idle mode (with the functional clock inactive) and a possible start condition is detected on the I2C line. The STC WakeUp is generated only if the I2C_SYSC.ClockActivity field indicates that the functional clock can be disabled. Note that if the functional clock is not active, the start condition is asynchronously detected (no filtering and synchronization is used). For this reason, it is possible that the signalized start condition to be a glitch. If the functional clock cannot be disabled (I2C_SYSC.ClockActivity = 10 or 11), the programmer should not enable this wakeup, since the module has other synchronously detected WakeUp event that might be used to exit from idle mode, only if the detected transfer is accessing the I2C module.
		0	Start condition wakeup disabled
		1	Start condition wakeup enabled
5	GC_WE		General call IRQ wakeup enable. This read/write bit is used to enable or disable wakeup signal generation when I2C module is in idle mode and a general call is received on I2C line.
		0	General call wakeup disabled
		1	General call wakeup enabled
4	Reserved	0	Reserved
3	DRDY_WE		Receive/Transmit data ready IRQ wakeup enable. This read/write bit is used to enable or disable wakeup signal generation when I2C module is involved into a long transfer and no more registers accesses are performed on the interface (for example module are set in F/S I2C master transmitter mode and FIFO is full). If in the middle of such a transaction, the FIFO buffer needs more data to be transferred, CPU must be informed to write (in case of transmitter mode) or read (if receiver mode) in/from the FIFO.
		0	Transmit/receive data ready wakeup disabled
		1	Transmit/receive data ready wakeup enabled
2	ARDY_WE		Register access ready IRQ wakeup enable. This read/write bit is used to enable or disable wakeup signal generation when I2C module is involved into a long transfer and no more registers accesses are performed on the interface (for example the module is set in F/S I2C master transmitter mode and FIFO is full). If the current transaction is finished, the module needs to inform CPU about transmission completion.
		0	Register access ready wakeup disabled
		1	Register access ready wakeup enabled
1	NACK_WE		No acknowledgment IRQ wakeup enable. This read/write bit is used to enable or disable wakeup signal generation when I2C module is involved into a long transfer and no more registers accesses are performed on the interface (for example the module is set in F/S I2C master transmitter mode and FIFO is full). If in the middle of such of a transaction a Not Acknowledgment event is raised, the module needs to inform CPU about transmission error.
		0	Not Acknowledge wakeup disabled
		1	Not Acknowledge wakeup enabled

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 20-17. I2C Wakeup Enable Register (I2C_WE) Field Descriptions (continued)

Bit	Field	Value	Description
0	AL_WE		Arbitration lost IRQ wakeup enable. This read/write bit is used to enable or disable wakeup signal generation when I2C module is configured as a master and it loses the arbitration. This wake up is very useful when the module is configured as a master transmitter, all the necessary data is provided in the FIFO Tx, STT is enabled and the module enters in Idle Mode. If the module loses the arbitration, an Arbitration Lost event is raised and the module needs to inform CPU about transmission error. **Note:** The AL wakeup must be enabled only for multimaster communication. If the AL_WE is not enabled and the scenario described above occurs, the module will not be able to inform the CPU about the state of the transfer and it will be blocked in an undetermined state.
		0	Arbitration lost wakeup disabled
		1	Arbitration lost wakeup enabled

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.4.10 Receive DMA Enable Set Register (I2C_DMARXENABLE_SET)

The 1-bit field enables a receive DMA request. Writing a 1 to this field will set it to 1. Writing a 0 will have no effect, that is, the register value is not modified. Note that the I2C_BUF.RDMA_EN field is the global (slave) DMA enabler, and that it is disabled by default. The I2C_BUF.RDMA_EN field should also be set to 1 to enable a receive DMA request.

Figure 20-25. Receive DMA Enable Set Register (I2C_DMARXENABLE_SET)

31	1	0
Reserved		DMARX_ENABLE_SET
R-0		R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20-18. Receive DMA Enable Set Register (I2C_DMARXENABLE_SET) Field Descriptions

Bit	Field	Value	Description
31-1	Reserved	0	Reserved
0	DMARX_ENABLE_SET	0-1	Receive DMA channel enable set.

20.4.11 Transmit DMA Enable Set Register (I2C_DMATXENABLE_SET)

The 1-bit field enables a transmit DMA request. Writing a 1 to this field will set it to 1. Writing a 0 will have no effect, that is, the register value is not modified. Note that the I2C_BUF.XDMA_EN field is the global (slave) DMA enabler, and that it is disabled by default. The I2C_BUF.XDMA_EN field should also be set to 1 to enable a transmit DMA request.

Figure 20-26. Receive DMA Enable Set Register (I2C_DMATXENABLE_SET)

31	1	0
Reserved		DMATX_ENABLE_SET
R-0		R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20-19. Transmit DMA Enable Set Register (I2C_DMATXENABLE_SET) Field Descriptions

Bit	Field	Value	Description
31-1	Reserved	0	Reserved
0	DMATX_TRANSMIT_SET	0-1	Transmit DMA channel enable set.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.4.12 Receive DMA Enable Clear Register (I2C_DMARXENABLE_CLR)

The 1-bit field disables a receive DMA request. Writing a 1 to a bit will clear it to 0. Another result of setting to ‘1’ the DMARX_ENABLE_CLEAR field, is the reset of the DMA RX request and wakeup lines. Writing a 0 will have no effect, that is, the register value is not modified.

Figure 20-27. Receive DMA Enable Set Register (I2C_DMARXENABLE_CLR)

31	1	0
Reserved		DMARX_ENABLE_CLEAR
R-0		R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20-20. Receive DMA Enable Clear Register (I2C_DMARXENABLE_CLR) Field Descriptions

Bit	Field	Value	Description
31-1	Reserved	0	Reserved
0	DMARX_ENABLE_CLEAR	0-1	Receive DMA channel enable clear.

20.4.13 Transmit DMA Enable Clear Register (I2C_DMATXENABLE_CLR)

The 1-bit field disables a transmit DMA request. Writing a 1 to a bit will clear it to 0. Another result of setting to ‘1’ the DMATX_ENABLE_CLEAR field, is the reset of the DMA TX request and wakeup lines. Writing a 0 will have no effect, that is, the register value is not modified.

Figure 20-28. Receive DMA Enable Set Register (I2C_DMATXENABLE_CLR)

31	1	0
Reserved		DMARX_ENABLE_CLEAR
R-0		R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20-21. Transmit DMA Enable Clear Register (I2C_DMATXENABLE_CLR) Field Descriptions

Bit	Field	Value	Description
31-1	Reserved	0	Reserved
0	DMARX_ENABLE_CLEAR	0-1	Receive DMA channel enable clear.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.4.14 Receive DMA Wakeup Register (I2C_DMARXWAKE_EN)

All 1-bit fields enable a specific (synchronous) DMA request source to generate an asynchronous wakeup (on the appropriate swakeup line). Note that the I2C_SYSC.ENAWAKEUP field is the global (slave) wakeup enabler, and that it is disabled by default.

Figure 20-29. Receive DMA Wakeup Register (I2C_DMARXWAKE_EN)

31							16
Reserved							
R-0							
15	14	13	12	11	10	9	8
Reserved	XDR	RDR	Reserved	ROVR	XUDF	AAS	BF
R-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
Reserved	STC	GC	Reserved	DRDY	ARDY	NACK	AL
R-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20-22. Receive DMA Wakeup Register (I2C_DMARXWAKE_EN) Field Descriptions

Bit	Field	Value	Description
31-15	Reserved	0	Reserved
14	XDR		Transmit draining wakeup set.
		0	Transmit draining interrupt disabled
		1	Transmit draining interrupt enabled
13	RDR		Receive draining wakeup set.
		0	Receive draining interrupt disabled
		1	Receive draining interrupt enabled
12	Reserved	0	Reserved
11	ROVR		Receive overrun wakeup set.
		0	Receive overrun interrupt disabled
		1	Receive draining interrupt enabled
10	XUDF		Transmit underflow wakeup set.
		0	Transmit underflow interrupt disabled
		1	Transmit underflow interrupt enabled
9	AAS		Address as slave IRQ wakeup set.
		0	Addressed as slave interrupt disabled
		1	Addressed as slave interrupt enabled
8	BF		Bus free IRQ wakeup set.
		0	Bus free wakeup disabled
		1	Bus free wakeup enabled
7	Reserved	0	Reserved
6	STC		Start condition IRQ wakeup set.
		0	Start condition wakeup disabled
		1	Start condition wakeup enabled
5	GC		General call IRQ wakeup set.
		0	General call wakeup disabled
		1	General call wakeup enabled
4	Reserved	0	Reserved
3	DRDY		Receive/transmit data ready IRQ wakeup set.
		0	Transmit/receive data ready wakeup disabled
		1	Transmit/receive data ready wakeup enabled

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 20-22. Receive DMA Wakeup Register (I2C_DMARXWAKE_EN) Field Descriptions (continued)

Bit	Field	Value	Description
2	ARDY		Register access ready IRQ wakeup set.
		0	Register access ready wakeup disabled
		1	Register access ready wakeup enabled
1	NACK		No acknowledgment IRQ wakeup set.
		0	Not Acknowledge wakeup disabled
		1	Not Acknowledge wakeup enabled
0	AL		Arbitration lost IRQ wakeup set.
		0	Arbitration lost wakeup disabled
		1	Arbitration lost wakeup enabled

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.4.15 Transmit DMA Wakeup Register (I2C_DMATXWAKE_EN)

All 1-bit fields enable a specific (synchronous) DMA request source to generate an asynchronous wakeup (on the appropriate swakeup line). Note that the I2C_SYSC.ENAWAKEUP field is the global (slave) wakeup enabler, and that it is disabled by default.

Figure 20-30. Receive DMA Wakeup Register (I2C_DMATXWAKE_EN)

31							16
Reserved							
R-0							
15	14	13	12	11	10	9	8
Reserved	XDR	RDR	Reserved	ROVR	XUDF	AAS	BF
R-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
Reserved	STC	GC	Reserved	DRDY	ARDY	NACK	AL
R-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20-23. Transmit DMA Wakeup Register (I2C_DMATXWAKE_EN) Field Descriptions

Bit	Field	Value	Description
31-15	Reserved	0	Reserved
14	XDR		Transmit draining wakeup set.
		0	Transmit draining interrupt disabled
		1	Transmit draining interrupt enabled
13	RDR		Receive draining wakeup set.
		0	Receive draining interrupt disabled
		1	Receive draining interrupt enabled
12	Reserved	0	Reserved
11	ROVR		Receive overrun wakeup set.
		0	Receive overrun interrupt disabled
		1	Receive draining interrupt enabled
10	XUDF		Transmit underflow wakeup set.
		0	Transmit underflow interrupt disabled
		1	Transmit underflow interrupt enabled
9	AAS		Address as slave IRQ wakeup set.
		0	Addressed as slave interrupt disabled
		1	Addressed as slave interrupt enabled
8	BF		Bus free IRQ wakeup set.
		0	Bus free wakeup disabled
		1	Bus free wakeup enabled
7	Reserved	0	Reserved
6	STC		Start condition IRQ wakeup set.
		0	Start condition wakeup disabled
		1	Start condition wakeup enabled
5	GC		General call IRQ wakeup set.
		0	General call wakeup disabled
		1	General call wakeup enabled
4	Reserved	0	Reserved
3	DRDY		Receive/transmit data ready IRQ wakeup set.
		0	Transmit/receive data ready wakeup disabled
		1	Transmit/receive data ready wakeup enabled

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 20-23. Transmit DMA Wakeup Register (I2C_DMATXWAKE_EN) Field Descriptions (continued)

Bit	Field	Value	Description
2	ARDY		Register access ready IRQ wakeup set.
		0	Register access ready wakeup disabled
		1	Register access ready wakeup enabled
1	NACK		No acknowledgment IRQ wakeup set.
		0	Not Acknowledge wakeup disabled
		1	Not Acknowledge wakeup enabled
0	AL		Arbitration lost IRQ wakeup set.
		0	Arbitration lost wakeup disabled
		1	Arbitration lost wakeup enabled

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.4.16 System Status Register (I2C_SYSS)

Figure 20-31. System Status Register (I2C_SYSS)

31	1	0
Reserved		RDONE
R-0		R/W-1

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20-24. System Status Register (I2C_SYSS) Field Descriptions

Bit	Field	Value	Description
31-1	Reserved	0	Reserved
0	RDONE		Reset done bit. This read-only bit indicates the state of the reset in case of hardware reset, global software reset (I2C_SYSC.SRST) or partial software reset (I2C_CON.I2C_EN). The module must receive all its clocks before it can grant a reset-completed status.
		0	Internal module reset in ongoing
		1	Reset completed
			Value after reset is low.

20.4.17 Buffer Configuration Register (I2C_BUF)

This read/write register enables DMA transfers and allows the configuration of FIFO thresholds for the FIFO management (see the FIFO Management subsection).

Figure 20-32. Buffer Configuration Register (I2C_BUF)

31						16
Reserved						
R-0						

15	14	13 8	7	6	5 0
RDMA_EN	RXFIFO_CLR	RXTRSH	XDMA_EN	TXFIFO_CLR	TXTRSH
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20-25. Buffer Configuration Register (I2C_BUF) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15	RDMA_EN		Receive DMA channel enable. When this bit is set to 1, the receive DMA channel is enabled and the receive data ready status bit (I2C_IRQSTATUS_RAW: RRDY) is forced to 0 by the core.
		0	Receive DMA channel disabled
		1	Receive DMA channel enabled
			Value after reset is low.
14	RXFIFO_CLR		Receive FIFO clear. When set, receive FIFO is cleared (hardware reset for RX FIFO generated). This bit is automatically reset by the hardware. During reads, it always returns 0.
		0	Normal mode
		1	Rx FIFO is reset
			Value after reset is low.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 20-25. Buffer Configuration Register (I2C_BUF) Field Descriptions (continued)

Bit	Field	Value	Description
13-8	RXTRSH	0-3Fh	Threshold value for FIFO buffer in RX mode. The receive threshold value is used to specify the trigger level for data receive transfers. The value is specified from the OCP point of view.
		0	Receive Threshold value = 1
		1	Receive Threshold value = 2
		-	
		-	
		3Fh	Receive Threshold value = 64
			Value after reset is 00h.
			For the FIFO management description, see the FIFO Management subsection.
			Note1: programmed threshold cannot exceed the actual depth of the FIFO.
			Note2: the threshold must not be changed while a transfer is in progress (after STT was configured or after the module was addressed as a slave).
7	XDMA_EN		Transmit DMA channel enable. When this bit is set to 1, the transmit DMA channel is enabled and the transmit data ready status (I2C_IRQSTATUS_RAW: XRDY) bit is forced to 0 by the core.
		0	Transmit DMA channel disabled
		1	Transmit DMA channel enabled
			Value after reset is low.
6	TXFIFO_CLR		Transmit FIFO clear. When set, transmit FIFO is cleared (hardware reset for TX FIFO). This bit is automatically reset by the hardware. During reads, it always returns 0.
		0	Normal mode
		1	Tx FIFO is reset
			Value after reset is low.
5-0	TXTRSH	0-3Fh	Threshold value for FIFO buffer in TX mode. The Transmit Threshold value is used to specify the trigger level for data transfers. The value is specified from the OCP point of view.
		0	Transmit Threshold value = 1
		1	Transmit Threshold value = 2
		-	
		-	
		3Fh	Transmit Threshold value = 64
			Value after reset is 00h
			Note1: programmed threshold cannot exceed the actual depth of the FIFO.
			Note2: the threshold must not be changed while a transfer is in progress (after STT was configured or after the module was addressed as a slave).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.4.18 Data Counter Register (I2C_CNT)

CAUTION

During an active transfer phase (between STT has been set to 1 and receiving of ARDY), no modification must be done in this register. Changing it may result in an unpredictable behavior.

This read/write register is used to control the numbers of bytes in the I2C data payload.

Figure 20-33. Data Counter Register (I2C_CNT)

31 – 16	15 – 0
Reserved	DCOUNT
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20-26. Data Counter Register (I2C_CNT) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15-0	DCOUNT	0-FFFFh	Data count. I2C Master Mode only (receive or transmit; F/S). This 16-bit countdown counter decrements by 1 for every byte received or sent through the I2C interface. A write initializes DCOUNT to a saved initial value. A read returns the number of bytes that are yet to be received or sent. A read into DCOUNT returns the initial value only before a start condition and after a stop condition. When DCOUNT reaches 0, the core generates a stop condition if a stop condition was specified (I2C_CON.STP = 1) and the ARDY status flag is set to 1 in the I2C_IRQSTATUS_RAW register. Note that DCOUNT must not be reconfigured after I2C_CON.STT was enabled and before ARDY is received. **Note1:** In case of I2C mode of operation, if I2C_CON.STP = 0, then the I2C asserts SCL = 0 when DCOUNT reaches 0. The CPU can then reprogram DCOUNT to a new value and resume sending or receiving data with a new start condition (restart). This process repeats until the CPU sets to 1 the I2C_CON.STP bit. The ARDY flag is set each time DCOUNT reaches 0 and DCOUNT is reloaded to its initial value.
		0	Data counter = 65536 bytes (216)
		1	Data counter = 1 bytes
		-	
		-	
		FFFFh	Data counter = 65535 bytes (216 - 1) Values after reset are low (all 16 bits). **Note2:** Since for DCOUNT = 0, the transfer length is 65536, the module does not allow the possibility to initiate zero data bytes transfers.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.4.19 Data Access Register (I2C_DATA)

This register is the entry point for the local host to read data from or write data to the FIFO buffer.

Figure 20-34. Data Access Register (I2C_DATA)

31 – 8	7 – 0
Reserved	DATA
R-0	R/W-x

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20-27. Data Access Register (I2C_DATA) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-0	DATA	0-FFh	Transmit/Receive data FIFO endpoint. When read, this register contains the received I2C data. When written, this register contains the byte value to transmit over the I2C data. In SYSTEST loop back mode (I2C_SYSTEST: TMODE = 11), this register is also the entry/receive point for the data. Values after reset are unknown (all 8-bits). **Note:** A read access, when the buffer is empty, returns the previous read data value. A write access, when the buffer is full, is ignored. In both events, the FIFO pointers are not updated and an Access Error (AERR) Interrupt is generated.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.4.20 I2C Configuration Register (I2C_CON)

CAUTION

During an active transfer phase (between STT has been set to 1 and receiving of ARDY), no modification must be done in this register (except STP enable). Changing it may result in an unpredictable behavior.

Figure 20-35. I2C Configuration Register (I2C_CON)

31							16
Reserved							
R-0							
15	14	13	12	11	10	9	8
I2C_EN	Reserved	OPMODE		STB	MST	TRX	XSA
R/W-0	R-0	R/W-0		R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
XOA0	XOA01	XOA02	XOA3	Reserved		STP	STT
R/W-0	R/W-0	R/W-0	R/W-0	R-0		R/W-0	R/W-0

LEGEND: RR/W-0/W = Read/Write; R = Read only; -*n* = value after reset

Table 20-28. I2C Configuration Register (I2C_CON) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15	I2C_EN		I2C module enable. When this bit is cleared to 0, the I2C controller is not enabled and reset. When 0, receive and transmit FIFOs are cleared and all status bits are set to their default values. All configuration registers (I2C_IRQENABLE_SET, I2C_IRQWAKE_SET, I2C_BUF, I2C_CNT, I2C_CON, I2C_OA, I2C_SA, I2C_PSC, I2C_SCLL and I2C_SCLH) are not reset, they keep their initial values and can be accessed. The CPU must set this bit to 1 for normal operation.
		0	Controller in reset. FIFO are cleared and status bits are set to their default value.
		1	Module enabled
			Value after reset is low.
14	Reserved	0	Reserved
13-12	OPMODE		Operation mode selection. These two bits select module operation mode.
		0h	I2C Fast/Standard mode
		1h	Reserved
		2h	Reserved
		3h	Reserved
			Value after reset is 00.
11	STB		Start byte mode (I2C master mode only). The start byte mode bit is set to 1 by the CPU to configure the I2C in start byte mode (I2C_SA = 0000 0001). See the Philips I2C spec for more details [1].
		0	Normal mode
		1	Start byte mode
			Value after reset is low.
10	MST		Master/slave mode (I2C mode only). When this bit is cleared, the I2C controller is in the slave mode and the serial clock (SCL) is received from the master device. When this bit is set, the I2C controller is in the master mode and generates the serial clock.
			Note: This bit is automatically cleared at the end of the transfer on a detected stop condition, in case of arbitration lost or when the module is configured as a master but addressed as a slave by an external master.
		0	Slave mode
		1	Master mode
			Value after reset is low.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 20-28. I2C Configuration Register (I2C_CON) Field Descriptions (continued)

Bit	Field	Value	Description
9	TRX		Transmitter/receiver mode (i2C master mode only). When this bit is cleared, the I2C controller is in the receiver mode and data on data line SDA is shifted into the receiver FIFO and can be read from I2C_DATA register. When this bit is set, the I2C controller is in the transmitter mode and the data written in the transmitter FIFO via I2C_DATA is shifted out on data line SDA.
		0	Receiver mode
		1	Transmitter mode
			Value after reset is low.
			The operating modes are defined as follows:
			MST / **TRX** / **Operating Modes**
			0 / x / Slave receiver
			0 / x / Slave transmitter
			1 / 0 / Master receiver
			1 / 1 / Master transmitter
8	XSA		Expand slave address. (I2C mode only). When set, this bit expands the slave address to 10-bit.
		0	7-bit address mode
		1	10-bit address mode
			Value after reset is low.
7	XOA0		Expand own address 0. (I2C mode only). When set, this bit expands the base own address (OA0) to 10-bit.
		0	7-bit address mode
		1	10-bit address mode
			Value after reset is low.
6	XOA1		Expand own address 1. (I2C mode only). When set, this bit expands the first alternative own address (OA1) to 10-bit.
		0	7-bit address mode
		1	10-bit address mode
			Value after reset is low.
5	XOA2		Expand own address 2. (I2C mode only). When set, this bit expands the second alternative own address (OA2) to 10-bit.
		0	7-bit address mode. (I2C mode only).
		1	10-bit address mode
			Value after reset is low.
4	XOA3		Expand own address 3. When set, this bit expands the third alternative own address (OA3) to 10-bit.
		0	7-bit address mode
		1	10-bit address mode
			Value after reset is low.
3-2	Reserved	0	Reserved
1	STP		Stop condition (I2C master mode only). This bit can be set to a 1 by the CPU to generate a stop condition. It is reset to 0 by the hardware after the stop condition has been generated. The stop condition is generated when DCOUNT passes 0.
			When this bit is not set to 1 before the end of the transfer (DCOUNT = 0), the stop condition is not generated and the SCL line is hold to 0 by the master, which can re-start a new transfer by setting the STT bit to 1.
		0	No action or stop condition detected
		1	Stop condition queried
			Value after reset is low

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 20-28. I2C Configuration Register (I2C_CON) Field Descriptions (continued)

Bit	Field	Value	Description
0	STT		Start condition (I2C master mode only). This bit can be set to a 1 by the CPU to generate a start condition. It is reset to 0 by the hardware after the start condition has been generated. The start/stop bits can be configured to generate different transfer formats.
		0	No action or start condition detected
		1	Start condition queried
			Value after reset is low.

STT	STP	Conditions	Bus Activities
1	0	Start	S-A-D
0	1	Stop	P
1	1	Start-Stop (DCOUNT= n)	S-A-D..(n)..D-P
1	0	Start (DCOUNT= n)	S-A-D..(n)..D

Note: DCOUNT is data count value in I2C_CNT register.

20.4.21 I2C Own Address Register (I2C_OA)

CAUTION

During an active transfer phase (between STT has been set to 1 and receiving of ARDY), no modification must be done in this register. Changing it may result in an unpredictable behavior.

This register is used to specify the module's base I2C 7-bit or 10-bit address (base own address).

Figure 20-36. I2C Own Address Register (I2C_OA)

31 – 10	9 – 0
Reserved	OA
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 20-29. I2C Own Address Register (I2C_OA) Field Descriptions

Bit	Field	Value	Description
31-10	Reserved	0	Reserved
9-0	OA	0-3FFh	Own address. This field specifies either: • A 10-bit address coded on OA [9:0] when XOA (Expand Own Address, I2C_CON[7]) is set to 1. • A 7-bit address coded on OA [6:0] when XOA (Expand Own Address, I2C_CON[7]) is cleared to 0. In this case, OA [9:7] bits must be cleared to 000 by application software. Value after reset is low (all 10 bits).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.4.22 I2C Slave Address Register (I2C_SA)

CAUTION

During an active transfer phase (between STT has been set to 1 and receiving of ARDY), no modification must be done in this register. Changing it may result in an unpredictable behavior.

This register is used to specify the addressed I2C module 7-bit or 10-bit address (slave address).

Figure 20-37. I2C Own Address Register (I2C_SA)

31 – 10	9 – 0
Reserved	SA
R-0	R/W-3FFh

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20-30. I2C Slave Address Register (I2C_SA) Field Descriptions

Bit	Field	Value	Description
31-10	Reserved	0	Reserved
9-0	SA	0-3FFh	Slave address. This field specifies either: • A 10-bit address coded on SA [9:0] when XSA (Expand Slave Address, I2C_CON[8]) is set to 1. • A 7-bit address coded on SA [6:0] when XSA (Expand Slave Address, I2C_CON[8]) is cleared to 0. In this case, SA [9:7] bits must be cleared to 000 by application software. Value after reset is low (all 10 bits).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.4.23 I2C Clock Prescaler Register (I2C_PSC)

CAUTION

During an active mode (I2C_EN bit in I2C_CON register is set to 1), no modification must be done in this register. Changing it may result in an unpredictable behavior.

This register is used to specify the internal clocking of the I2C peripheral core.

Figure 20-38. I2C Own Address Register (I2C_PSC)

31 – 8	7 – 0
Reserved	PSC
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20-31. I2C Clock Prescaler Register (I2C_PSC) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-0	PSC	0-FFh	Fast/Standard mode prescale sampling clock divider value. The core uses this 8-bit value to divide the system clock (SCLK) and generates its own internal sampling clock (ICLK) for Fast and Standard operation modes. The core logic is sampled at the clock rate of the system clock for the module divided by (PSC + 1).
		0h	Divide by 1
		1h	Divide by 2
		-	
		-	
		FFh	Divide by 256
			Value after reset is low (all 8 bits).

Figure 20-39. Clock Divider

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.4.24 I2C SCL Low Time Register (I2C_SCLL)

CAUTION

During an active mode (I2C_EN bit in I2C_CON register is set to 1), no modification must be done in this register. Changing it may result in an unpredictable behavior.

This register is used to determine the SCL low time value when master.

Figure 20-40. I2C SCL Low Time Register (I2C_SCLL)

31	8	7	0
Reserved		SLLC	
R-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20-32. I2C SCL Low Time Register (I2C_SCLL) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-0	SCLL	0-FFh	Fast/Standard mode SCL low time. I2C master mode only, (FS). This 8-bit value is used to generate the SCL low time value (tLOW) when the peripheral is operated in master mode. t_{LOW} = (SCLL + 7) * I_{CLK} time period, Value after reset is low (all 8 bits).

20.4.25 I2C SCL High Time Register (I2C_SCLH)

CAUTION

During an active mode (I2C_EN bit in I2C_CON register is set to 1), no modification must be done in this register. Changing it may result in an unpredictable behavior.

This register is used to determine the SCL high time value when master.

Figure 20-41. I2C SCL High Time Register (I2C_SCLH)

31	8	7	0
Reserved		SCLH	
R-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20-33. I2C SCL High Time Register (I2C_SCLH) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-0	SCLH	0-FFh	Fast/Standard mode SCL low time. I2C master mode only, (FS). This 8-bit value is used to generate the SCL high time value (tHIGH) when the peripheral is operated in master mode. - t_{HIGH} = (SCLH + 5) * I_{CLK} time period. Value after reset is low (all 8 bits).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.4.26 System Test Register (I2C_SYSTEST)

CAUTION

Never enable this register for normal I2C operation

This register is used to facilitate system-level tests by overriding some of the standard functional features of the peripheral. It allows testing of SCL counters, controlling the signals that connect to I/O pins, or creating digital loop-back for self-test when the module is configured in system test (SYSTEST) mode. It also provides stop/non-stop functionality in the debug mode.

Figure 20-42. System Test Register (I2C_SYSTEST)

31							16
Reserved							
R-0							
15	14	13	12	11	10	9	8
ST_EN	FREE	TMODE		SSB	Reserved		SCL_I_FUNC
R/W-0	R/W-0	R/W-0		R/W-0	R-0		R-1
7	6	5	4	3	2	1	0
SCL_O_FUNC	SDA_I_FUNC	SDA_O_FUNC	Reserved	SCL_I	SCL_O	SDA_I	SDA_O
R-1	R-1	R-1	R-0	R-0	R-0	R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20-34. System Test Register (I2C_SYSTEST) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15	ST_EN		System test enable. This bit must be set to 1 to permit other system test registers bits to be set.
		0	Normal mode. All others bits in register are read only.
		1	System test enabled. Permit other system test registers bits to be set.
			Value after reset is low.
14	FREE		Free running mode (on breakpoint). This bit is used to determine the state of the I2C controller when a breakpoint is encountered in the HLL debugger.
			Note: This bit can be set independently of ST_EN value. FREE = 0: the I2C controller stops immediately after completion of the on-going bit transfer. Stopping the transfer is achieved by forcing the SCL line low. Note that in this case there will be no status register updates.
			FREE = 1: the I2C interface runs free. When Suspend indication will be asserted, there will be no accesses on the OCP Interface (the CPU is in debug mode) and consequently the FIFOs will reach full/empty state (according to RX or TX modes) and the I2C SDA line will be kept low. Note that the status registers will be updated, but no DMA, IRQ or WakeUp will be generated.
			The status registers likely to be updated in this mode are: I2C_IRQSTATUS_RAW.XRDY, I2C_IRQSTATUS_RAW.RRDY, I2C_IRQSTATUS_RAW.XUDF, I2C_IRQSTATUS_RAW.ROVR, I2C_IRQSTATUS_RAW.ARDY and I2C_IRQSTATUS_RAW.NACK.
		0	Stop mode (on breakpoint condition). If Master mode, it stops after completion of the on-going bit transfer. In slave mode, it stops during the phase transfer when 1 byte is completely transmitted/received.
		1	Free running mode
			Value after reset is low.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 20-34. System Test Register (I2C_SYSTEST) Field Descriptions (continued)

Bit	Field	Value	Description
13-12	TMODE		Test mode select. In normal functional mode (ST_EN = 0), these bits are don't care. They are always read as 00 and a write is ignored. In system test mode (ST_EN = 1), these bits can be set according to the following table to permit various system tests.
		0	Functional mode (default)
		1h	Reserved
		2h	Test of SCL counters (SCLL, SCLH, PSC). SCL provides a permanent clock with master mode.
		3h	Loop back mode select + SDA/SCL IO mode select
			Values after reset are low (2 bits).
			SCL counter test mode: in this mode, the SCL pin is driven with a permanent clock as if mastered with the parameters set in the I2C_PSC, I2C_SCLL, and I2C_SCLH registers.
			Loop back mode: in the master transmit mode only, data transmitted out of the I2C_DATA register (write action) is received in the same I2C_DATA register via an internal path through the FIFO buffer. The DMA and interrupt requests are normally generated if enabled.
			SDA/SCL IO mode: in this mode, the SCL IO and SDA IO are controlled via the I2C_SYSTEST [5:0] register bits.
11	SSB		Set status bits. Writing 1 into this bit also sets the 6 read/clear-only status bits contained in I2C_IRQSTATUS_RAW register (bits 5:0) to 1. Writing 0 into this bit doesn't clear status bits that are already set; only writing 1 into a set status bit can clear it (see I2C_IRQSTATUS_RAW operation). This bit must be cleared prior attempting to clear a status bit.
		0	No action
		1	Set all interrupt status bits to 1
			Value after reset is low.
10-9	Reserved	0	Reserved
8	SCL_I_FUNC		SCL line input value (functional mode). This read-only bit returns the logical state taken by the SCL line (either 1 or 0). It is active both in functional and test mode
		Read:0	Read 0 from SCL line
		Read:1	Read 1 from SCL line
			Value after reset is low.
7	SCL_O_FUNC		SCL line output value (functional mode). This read-only bit returns the value driven by the module on the SCL line (either 1 or 0). It is active both in functional and test mode.
		Read:0	Driven 0 on SCL line
		Read:1	Driven 1 on SCL line
			Value after reset is low.
6	SDA_I_FUNC		SDA line input value (functional mode). This read-only bit returns the logical state taken by the SDA line (either 1 or 0). It is active both in functional and test mode.
		Read:0	Read 0 from SDA line
		Read:1	Read 1 from SDA line
			Value after reset is low.
5	SDA_O_FUNC		SDA line output value (functional mode). This read-only bit returns the value driven by the module on the SDA line (either 1 or 0). It is active both in functional and test mode.
		Read:0	Driven 0 to SDA line
		Read:1	Driven 1 to SDA line
			Value after reset is low.
4	Reserved	0	Reserved
3	SCL_I		SCL line sense input value. In normal functional mode (ST_EN = 0), this read-only bit always reads 0. In system test mode (ST_EN = 1 & TMODE = 11), this read-only bit returns the logical state taken by the SCL line (either 1 or 0).
		Read:0	Read 0 from SCL line
		Read:1	Read 1 from SCL line
			Value after reset is low.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 20-34. System Test Register (I2C_SYSTEST) Field Descriptions (continued)

Bit	Field	Value	Description
2	SCL_O		SCL line drive output value. In normal functional mode (ST_EN = 0), this bit is don't care. It always reads 0 and a write is ignored. In system test mode (ST_EN = 1 & TMODE = 11), a 0 forces a low level on the SCL line and a 1 puts the I2C output driver to a high-impedance state.
		Read:0	Forces 0 on the SCL data line
		Read:1	SCL output driver in high-impedance state
			Value after reset is low.
1	SDA_I		SDA line sense input value. In normal functional mode (ST_EN = 0), this read-only bit always reads 0. In system test mode (ST_EN = 1 & TMODE = 11), this read-only bit returns the logical state taken by the SDA line (either 1 or 0).
		Read:0	Read 0 from SDA line
		Read:1	Read 1 from SDA line
			Value after reset is low.
0	SDA_O		SDA line drive output value. In normal functional mode (ST_EN = 0), this bit is don't care. It reads as 0 and a write is ignored. In system test mode (ST_EN = 1 & TMODE = 11), a 0 forces a low level on the SDA line and a 1 puts the I2C output driver to a high-impedance state.
		0	Write 0 to SDA line
		1	Write 1 to SDA line
			Value after reset is low.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.4.27 I2C Buffer Status Register (I2C_BUFSTAT)

This read-only register reflects the status of the internal buffers for the FIFO management (see the FIFO Management subsection).

Figure 20-43. I2C Buffer Status Register (I2C_BUFSTAT)

31															16
Reserved															
R-0															

15	14	13							8	7	6	5							0
FIFODEPTH		RXSTAT								Reserved		TXSTAT							
R-0		R-0								R-0		R-0							

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20-35. I2C Buffer Status Register (I2C_BUFSTAT) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15-14	FIFODEPTH		Internal FIFO buffers depth. This read-only bit indicates the internal FIFO buffer depth.
		0	8-bytes FIFO
		1h	16-bytes FIFO
		2h	32-bytes FIFO
		3h	64-bytes FIFO
			Value after reset is given by the boundary module generic parameter.
13-8	RXSTAT	0-3Fh	RX buffer status. This read-only field indicates the number of bytes to be transferred from the FIFO at the end of the I2C transfer (when RDR is asserted). It corresponds to the level indication of the RX FIFO (number of written locations).
			Value after reset is 0.
7-6	Reserved	0	Reserved
5-0	TXSTAT	0-3Fh	TX buffer status. This read-only field indicates the number of data bytes still left to be written in the TX FIFO (it's equal with the initial value of I2C_CNT.DCOUNT minus the number of data bytes already written in the TX FIFO through the OCP Interface).
			Value after reset is equal with 0.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.4.28 Own Address 1 (OA1) (I2C_OA1)

CAUTION

During an active transfer phase (between STT has been set to 1 and receiving of ARDY), no modification must be done in this register. Changing it may result in an unpredictable behavior.

This register is used to specify the first alternative I2C 7-bit or 10-bit address (own address 1 - OA1).

Figure 20-44. Own Address 1 (OA1) (I2C_OA1)

31 – 10	9 – 0
Reserved	0A1
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20-36. Own Address 1 (OA1) (I2C_OA1) Field Descriptions

Bit	Field	Value	Description
31-10	Reserved	0	Reserved
9-0	OA1	0-3FFh	Own address 1. This field specifies either: • A 10-bit address coded on OA1 [9:0] when XOA1 (Expand Own Address 1 - XOA1, I2C_CON[6]) is set to 1. • A 7-bit address coded on OA1 [6:0] when XOA1 (Expand Own Address 1 – XOA1, I2C_CON[6]) is cleared to 0. In this case, OA1 [9:7] bits must be cleared to 000 by application software. Value after reset is low (all 10 bits).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.4.29 I2C Own Address 2 Register (I2C_OA2)

CAUTION

During an active transfer phase (between STT has been set to 1 and receiving of ARDY), no modification must be done in this register. Changing it may result in an unpredictable behavior.

This register is used to specify the first alternative I2C 7-bit or 10-bit address (own address 2 - OA2).

Figure 20-45. I2C Own Address 2 Register (I2C_OA2)

31 – 10	9 – 0
Reserved	0A2
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20-37. I2C Own Address 2 Register (I2C_OA2) Field Descriptions

Bit	Field	Value	Description
31-10	Reserved	0	Reserved
9-0	OA2	0-3FFh	Own address 2. This field specifies either: • A 10-bit address coded on OA2 [9:0] when XOA1 (Expand Own Address 2 - XOA2, I2C_CON[5]) is set to 1. • A 7-bit address coded on OA2 [6:0] when XOA2 (Expand Own Address 2 – XOA2, I2C_CON[5]) is cleared to 0. In this case, OA2 [9:7] bits must be cleared to 000 by application software. Value after reset is low (all 10 bits).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.4.30 I2C Own Address 3 Register (I2C_OA3)

CAUTION

During an active transfer phase (between STT has been set to 1 and receiving of ARDY), no modification must be done in this register. Changing it may result in an unpredictable behavior.

This register is used to specify the first alternative I2C 7-bit or 10-bit address (own address 3 - OA3).

Figure 20-46. I2C Own Address 3 Register (I2C_OA3)

31 – 10	9 – 0
Reserved	0A3
R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20-38. I2C Own Address 3 Register (I2C_OA3) Field Descriptions

Bit	Field	Value	Description
31-10	Reserved	0	Reserved
9-0	OA3	0-3FFh	Own address 2. This field specifies either: • A 10-bit address coded on OA3 [9:0] when XOA3 (Expand Own Address 3 - XOA3, I2C_CON[4]) is set to 1. • A 7-bit address coded on OA3 [6:0] when XOA1 (Expand Own Address 3 – XOA3, I2C_CON[4]) is cleared to 0. In this case, OA3 [9:7] bits must be cleared to 000 by application software. Value after reset is low (all 10 bits).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.4.31 Active Own Address Register (I2C_ACTOA)

This read-only register is used to indicate which one of the module's four own addresses the external master used when addressing the module. The CPU can read this register when the AAS indication was activated. The indication is cleared at the end of the transfer.

Figure 20-47. Active Own Address Register (I2C_ACTOA)

31					16
Reserved					
R-0					

15	4	3	2	1	0
Reserved		OA3_ACT	OA2_ACT	OA1_ACT	OA0_ACT
R-0		R-0	R-0	R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20-39. Active Own Address Register (I2C_ACTOA) Field Descriptions

Bit	Field	Value	Description
31-4	Reserved	0	Reserved
3	OA3_ACT		Own address 3 active. When a bit location is set to 1 by the core, it signalizes to the Local Host that an external master using the corresponding own address addressed the module.
		0	Own address inactive
		1	Own address active
			Value after reset is low.
2	OA2_ACT		Own address 2 active. When a bit location is set to 1 by the core, it signalizes to the Local Host that an external master using the corresponding own address addressed the module.
		0	Own address inactive
		1	Own address active
			Value after reset is low.
1	OA1_ACT		Own address 1 active. When a bit location is set to 1 by the core, it signalizes to the Local Host that an external master using the corresponding own address addressed the module.
		0	Own address inactive
		1	Own address active
			Value after reset is low.
0	OA0_ACT		Own address 0 active. When a bit location is set to 1 by the core, it signalizes to the Local Host that an external master using the corresponding own address addressed the module.
		0	Own address inactive
		1	Own address active
			Value after reset is low.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

20.4.32 I2C Clock Blocking Enable Register (I2C_SBLOCK)

This read/write register controls the automatic blocking of I2C clock feature in slave mode. It is used for the Local Host to configure for which of the 4 own addresses, the core must block the I2C clock (keep SCL line low) right after the Address Phase, when it is addressed as a slave.

Figure 20-48. I2C Clock Blocking Enable Register (I2C_SBLOCK)

31					16
Reserved					
R-0					

15	4	3	2	1	0
Reserved		OA3_EN	OA2_EN	OA1_EN	OA0_EN
R-0		R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 20-40. I2C Clock Blocking Enable Register (I2C_SBLOCK) Field Descriptions

Bit	Field	Value	Description
31-4	Reserved	0	Reserved
3	OA3_EN		Enable I2C clock blocking for own address 3. When the CPU sets a bit location to 1, if an external master using the corresponding own address addresses the core, the core will block the I2C clock right after the address phase. For releasing the I2C clock the CPU must write ‘0’ in the corresponding field.
		0	I2C clock released
		1	I2C clock blocked
			Value after reset is low.
2	OA2_EN		Enable I2C clock blocking for own address 2. When the CPU sets a bit location to 1, if an external master using the corresponding own address addresses the core, the core will block the I2C clock right after the address phase. For releasing the I2C clock the CPU must write ‘0’ in the corresponding field.
		0	I2C clock released
		1	I2C clock blocked
			Value after reset is low.
1	OA1_EN		Enable I2C clock blocking for own address 1. When the CPU sets a bit location to 1, if an external master using the corresponding own address addresses the core, the core will block the I2C clock right after the address phase. For releasing the I2C clock the CPU must write ‘0’ in the corresponding field.
		0	I2C clock released
		1	I2C clock blocked
			Value after reset is low.
0	OA0_EN		Enable I2C clock blocking for own address 0. When the CPU sets a bit location to 1, if an external master using the corresponding own address addresses the core, the core will block the I2C clock right after the address phase. For releasing the I2C clock the CPU must write ‘0’ in the corresponding field.
		0	I2C clock released
		1	I2C clock blocked
			Value after reset is low.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Chapter 21

SPRUH73–July 2011

Multichannel Audio Serial Port (McASP)

This chapter describes the McASP of the device.

Topic **Page**

21.1 Introduction **2670**
21.2 Integration **2672**
21.3 Functional Description **2674**
21.4 McASP Registers **2717**

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.1 Introduction

21.1.1 Purpose of the Peripheral

The multichannel audio serial port (McASP) functions as a general-purpose audio serial port optimized for the needs of multichannel audio applications. The McASP is useful for time-division multiplexed (TDM) stream, Inter-Integrated Sound (I2S) protocols, and intercomponent digital audio interface transmission (DIT). The McASP consists of transmit and receive sections that may operate synchronized, or completely independently with separate master clocks, bit clocks, and frame syncs, and using different transmit modes with different bit-stream formats. The McASP module also includes serializers that can be individually enabled to either transmit or receive.

21.1.2 Features

Features of the McASP include:

- Two independent clock generator modules for transmit and receive.
  - Clocking flexibility allows the McASP to receive and transmit at different rates. For example, the McASP can receive data at 48 kHz but output up-sampled data at 96 kHz or 192 kHz.
- Independent transmit and receive modules, each includes:
  - Programmable clock and frame sync generator.
  - TDM streams from 2 to 32, and 384 time slots.
  - Support for time slot sizes of 8, 12, 16, 20, 24, 28, and 32 bits.
  - Data formatter for bit manipulation.
- Individually assignable serial data pins (up to 6 pins).
- Glueless connection to audio analog-to-digital converters (ADC), digital-to-analog converters (DAC), codec, digital audio interface receiver (DIR), and S/PDIF transmit physical layer components.
- Wide variety of I2S and similar bit-stream format.
- Integrated digital audio interface transmitter (DIT) supports (up to 10 transmit pins):
  - S/PDIF, IEC60958-1, AES-3 formats.
  - Enhanced channel status/user data RAM.
- 384-slot TDM with external digital audio interface receiver (DIR) device.
  - For DIR reception, an external DIR receiver integrated circuit should be used with I2S output format and connected to the McASP receive section.
- Extensive error checking and recovery.
  - Transmit underruns and receiver overruns due to the system not meeting real-time requirements.
  - Early or late frame sync in TDM mode.
  - Out-of-range high-frequency master clock for both transmit and receive.
  - External error signal coming into the AMUTEIN input.
  - DMA error due to incorrect programming.

21.1.3 Protocols Supported

The McASP supports a wide variety of protocols.

- Transmit section supports:
  - Wide variety of I2S and similar bit-stream formats.
  - TDM streams from 2 to 32 time slots.
  - S/PDIF, IEC60958-1, AES-3 formats.
- Receive section supports:
  - Wide variety of I2S and similar bit-stream formats.
  - TDM streams from 2 to 32 time slots.
  - TDM stream of 384 time slots specifically designed for easy interface to external digital interface

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

receiver (DIR) device transmitting DIR frames to McASP using the I2S protocol (one time slot for each DIR subframe).

The transmit and receive sections may each be individually programmed to support the following options on the basic serial protocol:

- Programmable clock and frame sync polarity (rising or falling edge): ACLKR/X, AHCLKR/X, and AFSR/X.
- Slot length (number of bits per time slot): 8, 12, 16, 20, 24, 28, 32 bits supported.
- Word length (bits per word): 8, 12, 16, 20, 24, 28, 32 bits; always less than or equal to the time slot length.
- First-bit data delay: 0, 1, 2 bit clocks.
- Left/right alignment of word inside slot.
- Bit order: MSB first or LSB first.
- Bit mask/pad/rotate function.
  - Automatically aligns data internally in either Q31 or integer formats.
  - Automatically masks nonsignificant bits (sets to 0, 1, or extends value of another bit).

In DIT mode for McASP, additional features of the transmitter are:

- Transmit-only mode 384 time slots (subframe) per frame.
- Bi-phase encoded 3.3 V output.
- Support for consumer and professional applications.
- Channel status RAM (384 bits).
- User data RAM (384 bits).
- Separate valid bit (V) for subframe A, B.

In I2S mode, the transmit and receive sections can support simultaneous transfers on up to all serial data pins operating as 192 kHz stereo channels.

In DIT mode, the transmitter can support a 192 kHz frame rate (stereo) on up to all serial data pins simultaneously (note that the internal bit clock for DIT runs two times faster than the equivalent bit clock for I2S mode, due to the need to generate Biphase Mark Encoded Data).

21.1.4 Unsupported McASP Features

The unsupported McASP features in this device include the following:

- AXR[9:4] — Signals are not pinned out
- AMUTE — Not connected
- AMUTEIN — Not connected

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.2 Integration

The device contains two instantiations of the McASP subsystem: McASP0 and McASP1. The McASP subsystem includes a McASP peripheral, and transmit/receive buffers.

Each McASP is configured with 10 serializers but only serializers 0–3 are pinned out on this device.

Figure 21-1. McASP0–1 Integration

21.2.1 McASP Connectivity Attributes

The general connectivity attributes for the McASP modules are summarized in Table 21-1

Table 21-1. McASP Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_L3S_GCLK (OCP Clock) PD_PER_MCASP_FCLK (Aux Clock)
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	1 Transmit Interrupt x_intr_pend - to MPU subsystem and PRUSS 1 Receive Interrupt r_intr_pend - to MPU subsystem and PRUSS
DMA Requests	2 DMA requests to EDMA
Physical Address	L3 Slow slave port (data) L4 Peripheral slave port (CFG)

DRAFT

Copyright © 2011, Texas Instruments Incorporated

21.2.2 McASP Clock and Reset Management

The McASP module uses functional clocks either generated internally (master mode) or supplied from its serial interface (slave mode). The internal interconnect interface clock is used for the module internal OCP interface. Internal registers select the source of the functional clocks and the applied divider ratio.

Table 21-2. McASP Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
ocp_clk Interface clock	100 MHz	CORE_COLKOUTM4 / 2	pd_per_l3s_gclk From PRCM
auxclk Functional clock	24 MHz	CLK_M_OSC	pd_per_mcasp_gfclk From PRCM

21.2.3 McASP Pin List

The McASP external interface signals are shown in Table 21-3.

Table 21-3. McASP Pin List

Pin	Type	Description
McASP*x*_AXR[3:0]	I/O	Audio transmit/receive pin
McASP*x*_ACLKX	I/O	Transmit clock
McASP*x*_FSX	I/O	Frame synch for transmit
McASP*x*_AHCLKX	I/O	High speed transmit clock
McASP*x*_ACLKR	I/O	Receive clock
McASP*x*_FSR	I/O	Frame synch for receive
McASP*x*_AHCLKR	I/O	High speed receive clock

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.3 Functional Description

21.3.1 Overview

shows the major blocks of the McASP. The McASP has independent receive/transmit clock generators and frame sync generators, error-checking logic, and up to 6 serial data pins. Refer to the device-specific data manual for the number of data pins available on your device.

All the McASP pins on the device may be individually programmed as general-purpose I/O (GPIO) if they are not used for serial port functions.

The McASP includes the following pins:

- Serializers;
  - Data pins AXRn: Up to 6 pins.
- Transmit clock generator:
  - AHCLKX: McASP transmit high-frequency master clock.
  - ACLKX: McASP transmit bit clock.
- Transmit Frame Sync Generator;
  - AFSX: McASP transmit frame sync or left/right clock (LRCLK).
- Receive clock generator;
  - AHCLKR: McASP receive high-frequency master clock.
  - ACLKR: McASP receive bit clock.
- Receive Frame Sync Generator;
  - AFSR: McASP receive frame sync or left/right clock (LRCLK).
- Mute in/out;
  - AMUTEIN: McASP mute input (from external device).
  - AMUTE: McASP mute output.
  - Data pins AXRn.

21.3.2 Clock and Frame Sync Generators

The McASP clock generators are able to produce two independent clock zones: transmit and receive clock zones. The serial clock generators may be programmed independently for the transmit section and the receive section, and may be completely asynchronous to each other. The serial clock (clock at the bit rate) may be sourced:

- Internally - by passing through two clock dividers off the internal clock source (AUXCLK).
- Externally - directly from ACLKR/X pin.
- Mixed - an external high-frequency clock is input to the McASP on either the AHCLKX or AHCLKR pins, and divided down to produce the bit rate clock.

In the internal/mixed cases, the bit rate clock is generated internally and should be driven out on the ACLKX (for transmit) or ACLKR (for receive) pins. In the internal case, an internally-generated high-frequency clock may be driven out onto the AHCLKX or AHCLKR pins to serve as a reference clock for other components in the system.

The McASP requires a minimum of a bit clock and a frame sync to operate, and provides the capability to reference these clocks from an external high-frequency master clock. In DIT mode, it is possible to use only internally-generated clocks and frame syncs. Both the AUXCLK and System Clock are generated from SYSCLK2 (CLKDIV6 domain).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.3.2.1 Transmit Clock

The transmit bit clock, ACLKX, (Figure 21-2) may be either externally sourced from the ACLKX pin or internally generated, as selected by the CLKXM bit. If internally generated (CLKXM = 1), the clock is divided down by a programmable bit clock divider (CLKXDIV) from the transmit high-frequency master clock (AHCLKX).

Internally, the McASP always shifts transmit data at the rising edge of the internal transmit clock, XCLK, (Figure 21-2). The CLKXP mux determines if ACLKX needs to be inverted to become XCLK. If CLKXP = 0, the CLKXP mux directly passes ACLKX to XCLK. As a result, the McASP shifts transmit data at the rising edge of ACLKX. If CLKXP = 1, the CLKX mux passes the inverted version of ACLKX to XCLK. As a result, the McASP shifts transmit data at the falling edge of ACLKX.

The transmit high-frequency master clock, AHCLKX, may be either externally sourced from the AHCLKX pin or internally generated, as selected by the HCLKXM bit. If internally generated (HCLKXM = 1), the clock is divided down by a programmable high clock divider (HCLKXDIV) from McASP internal clock source AUXCLK. The transmit high-frequency master clock may be (but is not required to be) output on the AHCLKX pin where it is available to other devices in the system.

The transmit clock configuration is controlled by the following registers:

- ACLKXCTL.
- AHCLKXCTL.

Figure 21-2. Transmit Clock Generator Block Diagram

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.3.2.2 Receive Clock

The receiver also has the option to operate synchronously from the ACLKX and AFSX signals. This is achieved when the ASYNC bit in the transmit clock control register (ACLKXCTL) is cleared to 0 (see Figure 21-3). The receiver may be configured with different polarity (CLKRP) and frame sync data delay options from those options of the transmitter.

The receive clock configuration is controlled by the following registers:

- ACLKRCTL.
- AHCLKRCTL.

Figure 21-3. Receive Clock Generator Block Diagram

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.3.2.3 Frame Sync Generator

There are two different modes for frame sync: burst and TDM. A block diagram of the frame sync generator is shown in Figure 21-4. The frame sync options are programmed by the receive and transmit frame sync control registers (AFSRCTL and AFSXCTL). The options are:

- Internally-generated or externally-generated.
- Frame sync polarity: rising edge or falling edge.
- Frame sync width: single bit or single word.
- Bit delay: 0, 1, or 2 cycles before the first data bit.

The transmit frame sync pin is AFSX and the receive frame sync pin is AFSR. A typical usage for these pins is to carry the left/right clock (LRCLK) signal when transmitting and receiving stereo data.

Regardless if the AFSX/AFSR is internally generated or externally sourced, the polarity of AFSX/AFSR is determined by FSXP/FSRP, respectively, to be either rising or falling edge. If FSXP/FSRP = 0, the frame sync polarity is rising edge. If FSRP/FSRP = 1, the frame sync polarity is falling edge.

Figure 21-4. Frame Sync Generator Block Diagram

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.3.2.4 Clocking Examples

Some examples of processes using the McASP clocking and frame flexibility are:

- Receive data from a DVD at 48 kHz, but output up-sampled or decoded audio at 96 kHz or 192 kHz. This could be accomplished by inputting a high-frequency master clock (for example, 512 × receive FS), receiving with an internally-generated bit clock ratio of divide-by-8, and transmitting with an internally-generated bit clock ratio of divide-by-4 or divide-by-2.
- Transmit/receive data based on one sample rate (for example, 44.1 kHz), and transmit/receive data at a different sample rate (for example, 48 kHz).

21.3.3 Memory Map

See the device-specific data manual for information describing the device memory-map.

21.3.4 Signal Descriptions

The signals used on the McASP audio interface are listed in Table 21-4.

Table 21-4. McASP Interface Signals

Pin	I/O/Z	Device Reset (RESET = 0)	Description
		Transmitter Control	
AHCLKX	I/O/Z	Input	Transmit high-frequency master clock
AFSX	I/O/Z	Input	Transmit frame sync or left/right clock (LRCLK)
ACLKX	I/O/Z	Input	Transmit bit clock
		Receiver Control	
AHCLKR	I/O/Z	Input	Receive high-frequency master clock
AFSR	I/O/Z	Input	Receive frame sync or left/right clock (LRCLK)
ACLKR	I/O/Z	Input	Receive bit clock
		Mute	
AMUTE	I/O/Z	Input	Mute output
AMUTEIN	I/O/Z	Input	Mute input
		Data	
AXRn	I/O/Z	Input	TX/RX data pins

21.3.5 Pin Multiplexing

The McASP signals share pins with other processor functions. For detailed information on the McASP pin multiplexing and configuration, see the pin multiplexing information in the device-specific data manual.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.3.6 Transfer Modes

21.3.6.1 Burst Transfer Mode

The McASP supports a burst transfer mode, which is useful for nonaudio data such as passing control information between two processors. Burst transfer mode uses a synchronous serial format similar to the TDM mode. The frame sync generation is not periodic or time-driven as in TDM mode, but data driven, and the frame sync is generated for each data word transferred.

When operating in burst frame sync mode (Figure 21-5), as specified for transmit (XMOD = 0 in AFSXCTL) and receive (RMOD = 0 in AFSRCTL), one slot is shifted for each active edge of the frame sync signal that is recognized. Additional clocks after the slot and before the next frame sync edge are ignored.

In burst frame sync mode, the frame sync delay may be specified as 0, 1, or 2 serial clock cycles. This is the delay between the frame sync active edge and the start of the slot. The frame sync signal lasts for a single bit clock duration (FRWID = 0 in AFSRCTL, FXWID = 0 in AFSXCTL).

For transmit, when generating the transmit frame sync internally, the frame sync begins when the previous transmission has completed and when all the XBUF[n] (for every serializer set to operate as a transmitter) has been updated with new data.

For receive, when generating the receive frame sync internally, frame sync begins when the previous transmission has completed and when all the RBUF[n] (for every serializer set to operate as a receiver) has been read.

Figure 21-5. Burst Frame Sync Mode

The control registers must be configured as follows for the burst transfer mode. The burst mode specific bit fields are in bold face:

- PFUNC: The clock, frame, data pins must be configured for McASP function.
- PDIR: The clock, frame, data pins must be configured to the direction desired.
- PDOUT, PDIN, PDSET, PDCLR: Not applicable. Leave at default.
- GBLCTL: Follow the initialization sequence in Section 21.3.10.2 to configure this register.
- AMUTE: Not applicable. Leave at default.
- DLBCTL: If loopback mode is desired, configure this register according to Section 21.3.8.5, otherwise leave this register at default.
- DITCTL: DITEN must be left at default 0 to select non-DIT mode. Leave the register at default.
- RMASK/XMASK: Mask desired bits according to Section 21.3.7.2 and Section 21.3.8.3.
- RFMT/XFMT: Program all fields according to data format desired. See Section 21.3.8.3.
- AFSRCTL/AFSXCTL: Clear **RMOD/XMOD** bits to 0 to indicate burst mode. Clear **FRWID/FXWID** bits to 0 for single bit frame sync duration. Configure other fields as desired.
- ACLKRCTL/ACLKXCTL: Program all fields according to bit clock desired. See Section 21.3.2.
- AHCLKRCTL/AHCLKXCTL: Program all fields according to high-frequency clock desired. See Section 21.3.2.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

- RTDM/XTDM: Program RTDMS0/XTDMS0 to 1 to indicate one active slot only. Leave other fields at default.
- RINTCTL/XINTCTL: Program all fields according to interrupts desired.
- RCLKCHK/XCLKCHK: Not applicable. Leave at default.
- SRCTLn: Program SRMOD to inactive/transmitter/receiver as desired. DISMOD is not applicable and should be left at default.
- DITCSRA[n], DITCSRB[n], DITUDRA[n], DITUDRB[n]: Not applicable. Leave at default.

21.3.6.2 Time-Division Multiplexed (TDM) Transfer Mode

The McASP time-division multiplexed (TDM) transfer mode supports the TDM format discussed in .

Transmitting data in the TDM transfer mode requires a minimum set of pins:

- ACLKX - transmit bit clock.
- AFSX - transmit frame sync (or commonly called left/right clock).
- One or more serial data pins, AXRn, whose serializers have been configured to transmit.

The transmitter has the option to receive the ACLKX bit clock as an input, or to generate the ACLKX bit clock by dividing down the AHCLKX high-frequency master clock. The transmitter can either generate AHCLKX internally or receive AHCLKX as an input. See Section 21.3.2.1.

Similarly, to receive data in the TDM transfer mode requires a minimum set of pins:

- ACLKR - receive bit clock.
- AFSR - receive frame sync (or commonly called left/right clock).
- One or more serial data pins, AXRn, whose serializers have been configured to receive.

The receiver has the option to receive the ACLKR bit clock as an input or to generate the ACLKR bit clock by dividing down the AHCLKR high-frequency master clock. The receiver can either generate AHCLKR internally or receive AHCLKR as an input. See Section 21.3.2.2 and Section 21.3.2.3.

The control registers must be configured as follows for the TDM mode. The TDM mode specific bit fields are in bold face:

- PFUNC: The clock, frame, data pins must be configured for McASP function.
- PDIR: The clock, frame, data pins must be configured to the direction desired.
- PDOUT, PDIN, PDSET, PDCLR: Not applicable. Leave at default.
- GBLCTL: Follow the initialization sequence in Section 21.3.10.2 to configure this register.
- AMUTE: Program all fields according to mute control desired.
- DLBCTL: If loopback mode is desired, configure this register according to Section 21.3.8.5, otherwise leave this register at default.
- DITCTL: DITEN must be left at default 0 to select TDM mode. Leave the register at default.
- RMASK/XMASK: Mask desired bits according to Section 21.3.7.2 and Section 21.3.8.3.
- RFMT/XFMT: Program all fields according to data format desired. See Section 21.3.8.3.
- AFSRCTL/AFSXCTL: Set **RMOD/XMOD** bits to 2-32 for TDM mode. Configure other fields as desired.
- ACLKRCTL/ACLKXCTL: Program all fields according to bit clock desired. See Section 21.3.2.
- AHCLKRCTL/AHCLKXCTL: Program all fields according to high-frequency clock desired. See Section 21.3.2.
- RTDM/XTDM: Program all fields according to the time slot characteristics desired.
- RINTCTL/XINTCTL: Program all fields according to interrupts desired.
- RCLKCHK/XCLKCHK: Program all fields according to clock checking desired.
- SRCTLn: Program all fields according to serializer operation desired.
- DITCSRA[n], DITCSRB[n], DITUDRA[n], DITUDRB[n]: Not applicable. Leave at default.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.3.6.2.1 TDM Time Slots

TDM mode on the McASP can extend to support multiprocessor applications, with up to 32 time slots per frame. For each of the time slots, the McASP may be configured to participate or to be inactive by configuring XTDM and/or RTDM (this allows multiple processors to communicate on the same TDM serial bus).

The TDM sequencer (separate ones for transmit and receive) functions in this mode. The TDM sequencer counts the slots beginning with the frame sync. For each slot, the TDM sequencer checks the respective bit in either XTDM or RTDM to determine if the McASP should transmit/receive in that time slot.

If the transmit/receive bit is active, the McASP functions normally during that time slot; otherwise, the McASP is inactive during that time slot; no update to the buffer occurs, and no event is generated. Transmit pins are automatically set to a high-impedance state, 0, or 1 during that slot, as determined by bit DISMOD in SRCTL[n].

Figure 21-6 shows when the transmit DMA event AXEVT is generated. See Section 21.3.8.1.1 for details on data ready and the initialization period indication. The transmit DMA event for an active time slot (slot N) is generated during the previous time slot (slot N - 1), regardless if the previous time slot (slot N - 1) is active or inactive.

During an active transmit time slot (slot N), if the next time slot (slot N + 1) is configured to be active, the copy from XRBUF[n] to XRSR[n] generates the DMA event for time slot N + 1. If the next time slot (slot N + 1) is configured to be inactive, then the DMA event will be delayed to time slot M - 1. In this case, slot M is the next active time slot. The DMA event for time slot M is generated during the first bit time of slot M - 1.

The receive DMA request generation does not need this capability, since the receive DMA event is generated after data is received in the buffer (looks back in time). If a time slot is disabled, then no data is copied to the buffer for that time slot and no DMA event is generated.

Figure 21-6. Transmit DMA Event (AXEVT) Generation in TDM Time Slots

A See Section 21.3.10.2, step 7a.

Copyright © 2011, Texas Instruments Incorporated

21.3.6.2.2 Special 384 Slot TDM Mode for Connection to External DIR

The McASP receiver also supports a 384 time slot TDM mode (DIR mode), to support S/PDIF, AES-3, IEC-60958 receiver ICs whose natural block (block corresponds to McASP frame) size is 384 samples. The advantage to using the 384 time slot TDM mode is that interrupts may be generated synchronous to the S/PDIF, AES-3, IEC-60958, such as the last slot interrupt.

The receive TDM time slot register (RTDM) should be programmed to all 1s during reception of a DIR block. Other TDM functionalities (for example, inactive slots) are not supported (only the slot counter counts the 384 subframes in a block).

To receive data in the DIR mode, the following pins are typically needed:

- ACLKR - receive bit clock.
- AFSR - receive frame sync (or commonly called left/right clock). In this mode, AFSR should be connected to a DIR which outputs a start of block signal, instead of LRCLK.
- One or more serial data pins, AXRn, whose serializers have been configured to receive.

For this special DIR mode, the control registers can be configured just as for TDM mode, except set RMOD in AFSRCTL to 384 to receive 384 time slots.

21.3.6.3 Digital Audio Interface Transmit (DIT) Transfer Mode

In addition to the TDM and burst transfer modes, which are suitable for transmitting audio data between ICs inside the same system, the digital audio interface transmit (DIT) transfer mode of the McASP also supports transmission of audio data in the S/PDIF, AES-3, or IEC-60958 format. These formats are designed to carry audio data between different systems through an optical or coaxial cable. The DIT mode only applies to serializers configured as transmitters, not receivers. Refer to for a description of the S/PDIF format.

21.3.6.3.1 Transmit DIT Encoding

The McASP operation in DIT mode is basically identical to the 2 time slot TDM mode, but the data transmitted is output as a biphase mark encoded bit stream, with preamble, channel status, user data, validity, and parity automatically stuffed into the bit stream by the McASP. The McASP includes separate validity bits for even/odd subframes and two 384-bit RAM modules to hold channel status and user data bits.

The transmit TDM time slot register (XTDM) should be programmed to all 1s during DIT mode. TDM functionality is not supported in DIT mode, except that the TDM slot counter counts the DIT subframes.

To transmit data in the DIT mode, the following pins are typically needed:

- AHCLKX - transmit high-frequency master clock.
- One or more serial data pins, AXRn, whose serializers have been configured to transmit.

AHCLKX is optional (the internal clock source may be used instead), but if used as a reference, the processor provides a clock check circuit that continually monitors the AHCLKX input for stability.

If the McASP is configured to transmit in the DIT mode on more than one serial data pin, the bit streams on all pins will be synchronized. In addition, although they will carry unique audio data, they will carry the same channel status, user data, and validity information.

The actual 24-bit audio data must always be in bit positions 23-0 after passing through the first three stages of the transmit format unit.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

For left-aligned Q31 data, the following transmit format unit settings process the data into right aligned 24-bit audio data ready for transmission:

- XROT = 010 (rotate right by 8 bits).
- XRVRS = 0 (no bit reversal, LSB first).
- XMASK = FFFF FF00h-FFFF 0000h (depending upon whether 24, 23, 22, 21, 20, 19, 18, 17, or 16 valid audio data bits are present).
- XPAD = 00 (pad extra bits with 0).

For right-aligned data, the following transmit format unit settings process the data into right aligned 24-bit audio data ready for transmission:

- XROT = 000 (rotate right by 0 bits).
- XRVRS = 0 (no bit reversal, LSB first).
- XMASK = 00FF FFFFh to 0000 FFFFh (depending upon whether 24, 23, 22, 21, 20, 19, 18, 17, or 16 valid audio data bits are present).
- XPAD = 00 (pad extra bits with 0).

21.3.6.3.2 Transmit DIT Clock and Frame Sync Generation

The DIT transmitter only works in the following configuration:

- In transmit frame control register (AFSXCTL):
  - Internally-generated transmit frame sync, FSXM = 1.
  - Rising-edge frame sync, FSXP = 0.
  - Bit-width frame sync, FXWID = 0.
  - 384-slot TDM, XMOD = 1 1000 0000b.
- In transmit clock control register (ACLKXCTL), ASYNC = 1.
- In transmit bitstream format register (XFMT), XSSZ = 1111 (32-bit slot size).

All combinations of AHCLKX and ACLKX are supported.

This is a summary of the register configurations required for DIT mode. The DIT mode specific bit fields are in bold face:

- PFUNC: The data pins must be configured for McASP function. If AHCLKX is used, it must also be configured for McASP function.
- PDIR: The data pins must be configured as outputs. If AHCLKX is used as an input reference, it should be configured as input. If internal clock source AUXCLK is used as the reference clock, it may be output on the AHCLKX pin by configuring AHCLKX as an output.
- PDOUT, PDIN, PDSET, PDCLR: Not applicable for DIT operation. Leave at default.
- GBLCTL: Follow the initialization sequence in Section 21.3.10.2 to configure this register.
- AMUTE: Program all fields according to mute control desired.
- DLBCTL: Not applicable. Loopback is not supported for DIT mode. Leave at default.
- DITCTL: **DITEN** bit must be set to 1 to enable DIT mode. Configure other bits as desired.
- RMASK: Not applicable. Leave at default.
- RFMT: Not applicable. Leave at default.
- AFSRCTL: Not applicable. Leave at default.
- ACLKRCTL: Not applicable. Leave at default.
- AHCLKRCTL: Not applicable. Leave at default.
- RTDM: Not applicable. Leave at default.
- RINTCTL: Not applicable. Leave at default.
- RCLKCHK: Not applicable. Leave at default.
- **XMASK**: Mask desired bits according to the discussion in this section, depending upon left-aligned or right-aligned internal data.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

- **XFMT: XDATDLY** = 0. **XRVRS** = 0. **XPAD** = 0. **XPBIT** = default (not applicable). **XSSZ** = Fh (32-bit slot). XBUSEL = configured as desired. **XROT** bit is configured according to the discussion in this section, either 0 or 8-bit rotate.
- **AFSXCTL**: Configure the bits according to the discussion in this section.
- **ACLKXCTL**: **ASYNC** = 1. Program CLKXDIV bits to obtain the bit clock rate desired. Configure CLKXP and CLKXM bits as desired, because CLKX is not actually used in the DIT protocol.
- **AHCLKXCTL**: Program all fields according to high-frequency clock desired.
- **XTDM**: Set to FFFF FFFFh for all active slots for DIT transfers.
- XINTCTL: Program all fields according to interrupts desired.
- XCLKCHK: Program all fields according to clock checking desired.
- SRCTLn: Set **SRMOD** = 1 (transmitter) for the DIT pins. DISMOD field is don't care for DIT mode.
- **DITCSRA[n], DITCSRB[n]**: Program the channel status bits as desired.
- **DITUDRA[n], DITUDRB[n]**: Program the user data bits as desired.

21.3.6.3.3 DIT Channel Status and User Data Register Files

The channel status registers (DITCSRA*n* and DITCSRB*n*) and user data registers (DITUDRA*n* and DITUDRB*n*) are not double buffered. Typically the programmer uses one of the synchronizing interrupts, such as last slot, to create an event at a safe time so the register may be updated. In addition, the CPU reads the transmit TDM slot counter to determine which word of the register is being used.

It is a requirement that the software avoid writing to the word of user data and channel status that are being used to encode the current time slot; otherwise, it will be indeterminate whether the old or new data is used to encode the bitstream.

The DIT subframe format is defined in . The channel status information (C) and User Data (U) are defined in these DIT control registers:

- DITCSRA0 to DITCSRA5: The 192 bits in these six registers contain the channel status information for the LEFT channel within each frame.
- DITCSRB0 to DITCSRB5: The 192 bits in these six registers contain the channel status information for the RIGHT channel within each frame.
- DITUDRA0 to DITUDRA5: The 192 bits in these six registers contain the user data information for the LEFT channel within each frame.
- DITUDRB0 to DITUDRB5: The 192 bits in these six registers contain the user data information for the RIGHT channel within each frame.

The S/PDIF block format is shown in . There are 192 frames within a block (frame 0 to frame 191). Within each frame there are two subframes (subframe 1 and 2 for left and right channels, respectively). The channel status and user data information sent on each subframe is summarized in Table 21-5.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 21-5. Channel Status and User Data for Each DIT Block

Frame	Subframe	Preamble	Channel Status defined in:	User Data defined in:
Defined by DITCSRA0, DITCSRB0, DITUDRA0, DITUDRB0				
0	1 (L)	B	DITCSRA0[0]	DITUDRA0[0]
0	2 (R)	W	DITCSRB0[0]	DITUDRB0[0]
1	1 (L)	M	DITCSRA0[1]	DITUDRA0[1]
1	2 (R)	W	DITCSRB0[1]	DITUDRB0[1]
2	1 (L)	M	DITCSRA0[2]	DITUDRA0[2]
2	2 (R)	W	DITCSRB0[2]	DITUDRB0[2]
…	…	…	…	…
31	1 (L)	M	DITCSRA0[31]	DITUDRA0[31]
31	2 (R)	W	DITCSRB0[31]	DITUDRB0[31]
Defined by DITCSRA1, DITCSRB1, DITUDRA1, DITUDRB1				
32	1 (L)	M	DITCSRA1[0]	DITUDRA1[0]
32	2 (R)	W	DITCSRB1[0]	DITUDRB1[0]
…	…	…	…	…
63	1 (L)	M	DITCSRA1[31]	DITUDRA1[31]
63	2 (R)	W	DITCSRB1[31]	DITUDRB1[31]
Defined by DITCSRA2, DITCSRB2, DITUDRA2, DITUDRB2				
64	1 (L)	M	DITCSRA2[0]	DITUDRA2[0]
64	2 (R)	W	DITCSRB2[0]	DITUDRB2[0]
…	…	…	…	…
95	1 (L)	M	DITCSRA2[31]	DITUDRA2[31]
95	2 (R)	W	DITCSRB2[31]	DITUDRB2[31]
Defined by DITCSRA3, DITCSRB3, DITUDRA3, DITUDRB3				
96	1 (L)	M	DITCSRA3[0]	DITUDRA3[0]
96	2 (R)	W	DITCSRB3[0]	DITUDRB3[0]
…	…	…	…	…
127	1 (L)	M	DITCSRA3[31]	DITUDRA3[31]
127	2 (R)	W	DITCSRB3[31]	DITUDRB3[31]
Defined by DITCSRA4, DITCSRB4, DITUDRA4, DITUDRB4				
128	1 (L)	M	DITCSRA4[0]	DITUDRA4[0]
128	2 (R)	W	DITCSRB4[0]	DITUDRB4[0]
…	…	…	…	…
159	1 (L)	M	DITCSRA4[31]	DITUDRA4[31]
159	2 (R)	W	DITCSRB4[31]	DITUDRB4[31]
Defined by DITCSRA5, DITCSRB5, DITUDRA5, DITUDRB5				
160	1 (L)	M	DITCSRA5[0]	DITUDRA5[0]
160	2 (R)	W	DITCSRB5[0]	DITUDRB5[0]
…	…	…	…	…
191	1 (L)	M	DITCSRA5[31]	DITUDRA5[31]
191	2 (R)	W	DITCSRB5[31]	DITUDRB5[31]

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.3.7 General Architecture

21.3.7.1 Serializers

Figure 21-7 shows the block diagram of the serializer and its interface to other units within the McASP. The serializers take care of shifting serial data in and out of the McASP. Each serializer consists of a shift register (XRSR), a data buffer (XRBUF), a control register (SRCTL), and logic to support the data alignment options of the McASP. For each serializer, there is a dedicated serial data pin (AXRn) and a dedicated control register (SRCTL[n]). The control register allows the serializer to be configured as a transmitter, receiver, or as inactive. When configured as a transmitter the serializer shifts out data to the serial data pin AXRn. When configured as a receiver, the serializer shifts in data from the AXRn pin. The serializer is clocked from the transmit/receive section clock (ACLKX/ACLKR) if configured to transmit/receive respectively.

All serializers that are configured to transmit operate in lock-step. Similarly, all serializers that are configured to receive also operate in lock-step. This means that at most there are two zones per McASP, one for transmit and one for receive.

Figure 21-7. Individual Serializer and Connections Within McASP

For receive, data is shifted in through the AXRn pin to the shift register XRSR. Once the entire slot of data is collected in the XRSR, the data is copied to the data buffer XRBUF. The data is now ready to be read by the processor through the RBUF register, which is an alias of the XRBUF for receive. When the processor reads from the RBUF, the McASP passes the data from RBUF through the receive format unit and returns the formatted data to the processor.

For transmit, the processor services the McASP by writing data into the XBUF register, which is an alias of the XRBUF for transmit. The data automatically passes through the transmit format unit before actually reaching the XRBUF register in the serializer. The data is then copied from XRBUF to XRSR, and shifted out from the AXRn synchronously to the serial clock.

In DIT mode, in addition to the data, the serializer shifts out other DIT-specific information accordingly (preamble, user data, etc.).

The serializer configuration is controlled by SRCTL[n].

21.3.7.2 Format Unit

The McASP has two data formatting units, one for transmit and one for receive. These units automatically remap the data bits within the transmitted and received words between a natural format for the processor (such as a Q31 representation) and the required format for the external serial device (such as "I2S format"). During the remapping process, the format unit also can mask off certain bits or perform sign extension.

Since all transmitters share the same data formatting unit, the McASP only supports one transmit format at a time. For example, the McASP will not transmit in "I2S format" on serializer 0, while transmitting "Left Justified" on serializer 1. Likewise, the receiver section of the McASP only supports one data format at a time, and this format applies to all receiving serializers. However, the McASP can transmit in one format while receiving in a completely different format.

Copyright © 2011, Texas Instruments Incorporated

This formatting unit consists of three stages:

- Bit mask and pad (masks off bits, performs sign extension)
- Rotate right (aligns data within word)
- Bit reversal (selects between MSB first or LSB first)

Figure 21-8 shows a block diagram of the receive formatting unit, and Figure 21-9 shows the transmit formatting unit. Note that the order in which data flows through the three stages is different between the transmit and receive formatting units.

Figure 21-8. Receive Format Unit

Figure 21-9. Transmit Format Unit

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

The bit mask and pad stage includes a full 32-bit mask register, allowing selected individual bits to either pass through the stage unchanged, or be masked off. The bit mask and pad then pad the value of the masked off bits by inserting either a 0, a 1, or one of the original 32 bits as the pad value. The last option allows for sign-extension when the sign bit is selected to pad the remaining bits.

The rotate right stage performs bitwise rotation by a multiple of 4 bits (between 0 and 28 bits), programmable by the (R/X)FMT register. Note that this is a rotation process, not a shifting process, so bit 0 gets shifted back into bit 31 during the rotation.

The bit reversal stage either passes all 32 bits directly through, or swaps them. This allows for either MSB or LSB first data formats. If bit reversal is not enabled, then the McASP will naturally transmit and receive in an LSB first order.

Finally, note that the (R/X)DATDLY bits in (R/X)FMT also determine the data format. For example, the difference between I2S format and left-justified is determined by the delay between the frame sync edge and the first data bit of a given time slot. For I2S format, (R/X)DATDLY should be set to a 1-bit delay, whereas for left-justified format, it should be set to a 0-bit delay.

The combination of all the options in (R/X)FMT means that the McASP supports a wide variety of data formats, both on the serial data lines, and in the internal processor representation.

Section 21.3.8.3 provides more detail and specific examples. The examples use internal representation in integer and Q31 notation, but other fractional notations are also possible.

21.3.7.3 State Machine

The receive and transmit sections have independent state machines. Each state machine controls the interactions between the various units in the respective section. In addition, the state machine keeps track of error conditions and serial port status.

No serial transfers can occur until the respective state machine is released from reset. See initialization sequence for details (Section 21.3.10).

The receive state machine is controlled by the RFMT register, and it reports the McASP status and error conditions in the RSTAT register. Similarly, the transmit state machine is controlled by the XFMT register, and it reports the McASP status and error conditions in the XSTAT register.

21.3.7.4 TDM Sequencer

There are separate TDM sequencers for the transmit section and the receive section. Each TDM sequencer keeps track of the slot count. In addition, the TDM sequencer checks the bits of (R/X)TDM and determines if the McASP should receive/transmit in that time slot.

If the McASP should participate (transmit/receive bit is active) in the time slot, the McASP functions normally. If the McASP should not participate (transmit/receive bit is inactive) in the time slot, no transfers between the XRBUF and XRSR registers in the serializer would occur during that time slot. In addition, the serializers programmed as transmitters place their data output pins in a predetermined state (logic low, high, or high impedance) as programmed by each serializer control register (SRCTL). Refer also to Section 21.3.6.2 for details on how DMA event or interrupt generations are handled during inactive time slots in TDM mode.

The receive TDM sequencer is controlled by register RTDM and reports current receive slot to RSLOT. The transmit TDM sequencer is controlled by register XTDM and reports current transmit slot to XSLOT.

21.3.7.5 Clock Check Circuit

A common source of error in audio systems is a serial clock failure due to instabilities in the off-chip DIR circuit. To detect a clock error quickly, a clock-check circuit is included in the McASP for both transmit and receive clocks, since both may be sourced from off chip.

The clock check circuit can detect and recover from transmit and receive clock failures. See Section 21.3.8.4.6 for implementation and programming details.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.3.7.6 Pin Function Control

All McASP pins except AMUTEIN are bidirectional input/output pins. In addition, these bidirectional pins function either as McASP or general-purpose I/O (GPIO) pins. The following registers control the pin functions:

- Pin function register (PFUNC): selects pin to function as McASP or GPIO.
- Pin direction register (PDIR): selects pin to be input or output.
- Pin data input register (PDIN): shows data input at the pin.
- Pin data output register (PDOUT): data to be output at the pin if the pin is configured as GPIO output (PFUNC[n] = 1 and PDIR[n] = 1). Not applicable when the pin is configured as McASP pin (PFUNC[n] = 0).
- Pin data set register (PDSET): alias of PDOUT. Writing a 1 to PDSET[n] sets the respective PDOUT[n] to 1. Writing a 0 has no effect. Applicable only when the pin is configured as GPIO output (PFUNC[n] = 1 and PDIR[n] = 1).
- Pin data clear register (PDCLR): alias of PDOUT. Writing a 1 to PDCLR[n] clears the respective PDOUT[n] to 0. Writing a 0 has no effect. Applicable only when the pin is configured as GPIO output (PFUNC[n] = 1 and PDIR[n] = 1).

See the register descriptions in Section 21.4 for details on the mapping of each McASP pin to the register bits. Figure 21-10 shows the pin control block diagram.

Figure 21-10. McASP I/O Pin Control Block Diagram

21.3.7.6.1 McASP Pin Control-Transmit and Receive

You must correctly set the McASP GPIO registers PFUNC and PDIR, even when McASP pins are used for their serial port (non-GPIO) function.

Serial port functions include:

- Clock pins (ACLKX, ACLKR, AHCLKX, AHCLKR, AFSX, AFSR) used as clock inputs and outputs.
- Serializer data pins (AXRn) used to transmit or receive.
- AMUTE used as a mute output signal.

When using these pins in their serial port function, you must clear PFUNC[n] to 0 for each pin.

Also, certain outputs require PDIR[n] = 1, such as clock pins used as clock outputs, serializer data pins used to transmit, and AMUTE used as mute output.

Clock inputs and serializers configured to receive must have PDIR[n] = 0.

PFUNC and PDIR do not control the AMUTEIN signal, it is usually tied to a device level interrupt pin (consult device datasheet). If used as a mute input, this pin needs to be configured as an input in the appropriate peripheral.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Finally, there is an important advantage to having separate control of pin direction (by PDIR), and the choice of internal versus external clocking (by CLKRM/CLKXM). Depending on the specific device and usage, you might select an external clock (CLKRM = 0), while enabling the internal clock divider, and the clock pin as an output in the PDIR register (PDIR[ACLKR] = 1). In this case, the bit clock is an output (PDIR[ACLKR] = 1) and, therefore, routed to the ACLKR pin. However, because CLKRM = 0, the bit clock is then routed back to the McASP module as an "external" clock source. This may result in less skew between the clock inside the McASP and the clock in the external device, thus producing more balanced setup and hold times for a particular system. As a result, this may allow a higher serial clock rate interface.

21.3.8 Operation

This section discusses the operation of the McASP.

21.3.8.1 Data Transmission and Reception

The processor services the McASP by writing data to the XBUF register(s) for transmit operations, and by reading data from the RBUF register(s) for receive operations. The McASP sets status flag and notifies the processor whenever data is ready to be serviced. Section 21.3.8.1.1 discusses data ready status in detail.

The XBUF and RBUF registers can be accessed through one of the two peripheral ports of the device:

- The data port (DAT): This port is dedicated for data transfers on the device.
- The configuration bus (CFG): This port is used for both data transfers and peripheral configuration control on the device.

Section 21.3.8.1.2 and Section 21.3.8.1.3 discuss how to perform transfers through the data port and the configuration bus.

Either the CPU or the DMA can be used to service the McASP through any of these two peripheral ports. The CPU and DMA usages are discussed in Section 21.3.8.1.4 and Section 21.3.12.2.

21.3.8.1.1 Data Ready Status and Event/Interrupt Generation

21.3.8.1.1.1 Transmit Data Ready

The transmit data ready flag XDATA bit in the XSTAT register reflects the status of the XBUF register. The XDATA flag is set when data is transferred from the XRBUF[n] buffers to the XRSR[n] shift registers, indicating that the XBUF is empty and ready to accept new data from the processor. This flag is cleared when the XDATA bit is written with a 1, or when all the serializers configured as transmitters are written by the processor.

Whenever XDATA is set, an DMA event AXEVT is automatically generated to notify the DMA of the XBUF empty status. An interrupt AXINT is also generated if XDATA interrupt is enabled in the XINTCTL register (See Section 21.3.11.2 for details).

For DMA requests, the McASP does not require XSTAT to be read between DMA events. This means that even if XSTAT already has the XDATA flag set to 1 from a previous request, the next transfer triggers another DMA request.

Since all serializers act in lockstep, only one DMA event is generated to indicate that all active transmit serializers are ready to be written to with new data.

Figure 21-11 shows the timing details of when AXEVT is generated at the McASP boundary. In this example, as soon as the last bit (bit A0) of Word A is transmitted, the McASP sets the XDATA flag and generates an AXEVT event. However, it takes up to 5 McASP system clocks (AXEVT Latency) before AXEVT is active at the McASP boundary. Upon AXEVT, the processor can begin servicing the McASP by writing Word C into the XBUF (Processor Service Time). The processor must write Word C into the XBUF no later than the setup time required by the McASP (Setup Time).

The maximum Processor Service Time (Figure 21-11) can be calculated as:

Processor Service Time = Time Slot - AXEVT Latency - Setup Time

Copyright © 2011, Texas Instruments Incorporated

Figure 21-11. Processor Service Time Upon Transmit DMA Event (AXEVT)

A Refer to the device-specific data manual for the McASP system clock source. This is not the same as AUXCLK.

Example 21-1. Processor Service Time Calculation for Transmit DMA Event (AXEVT)

The following is an example to show how to calculate Processor Service Time. Assume the following setup:

- Device: C64x+ DSP at 300 MHz.
- McASP transmits in I2S format at 192 kHz frame rate. Assume slot size is 32 bit.

With the above setup, we obtain the following parameters corresponding to Figure 21-11:

- Calculation of McASP system clock cycle:
  - C64x+ DSP uses SYSCLK2 as the McASP system clock. It runs at 150 MHz (half of device frequency).
  - Therefore, McASP system clock cycle = 1/150 MHz = 6.7 ns.
- Calculation of ACLKX clock cycle:
  - This example has two 32-bit slots per frame, for a total of 64 bits per frame.
  - ACLKX clock cycle is (1/192 kHz)/64 = 81.4 ns.
- Time Slot between AXEVT events:
  - For I2S format, McASP generates two AXEVT events per 192 kHz frame.
  - Therefore, Time Slot between AXEVT events is (1/192 kHz)/2 = 2604 ns.
- AXEVT Latency:
  = 5 McASP system clocks
  = 6.7 ns × 5 = 33.5 ns
- Setup Time
  = 3 McASP system clocks + 4 ACLKX cycles
  = (6.7 ns × 3) + (81.4 ns × 4)
  = 345.7 ns
- Processor Service Time
  = Time Slot - AXEVT Latency - Setup Time
  = 2604 ns - 33.5 ns - 345.7 ns
  = 2225 ns

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.3.8.1.1.2 Receive Data Ready

Similarly, the receive data ready flag RDATA bit in the RSTAT reflects the status of the RBUF register. The RDATA flag is set when data is transferred from the XRSR[n] shift registers to the XRBUF[n] buffers, indicating that the RBUF contains received data and is ready to have the processor read the data. This flag is cleared when the RDATA bit is written with a 1, or when all the serializers configured as receivers are read.

Whenever RDATA is set, an DMA event AREVT is automatically generated to notify the DMA of the RBUF ready status. An interrupt ARINT is also generated if RDATA interrupt is enabled in the RINTCTL register (See Section 21.3.11.3 for details).

For DMA requests, the McASP does not require RSTAT to be read between DMA events. This means that even if RSTAT already has the RDATA flag set to 1 from a previous request, the next transfer triggers another DMA request.

Since all serializers act in lockstep, only one DMA event is generated to indicate that all active receive serializers are ready to receive new data.

Figure 21-12 shows the timing details of when AREVT is generated at the McASP boundary. In this example, as soon as the last bit (bit A0) of Word A is received, the McASP sets the RDATA flag and generates an AREVT event. However, it takes up to 5 McASP system clocks (AREVT Latency) before AREVT is active at the McASP boundary. Upon AREVT, the processor can begin servicing the McASP by reading Word A from the RBUF (Processor Service Time). The processor must read Word A from the XBUF no later than the setup time required by the McASP (Setup Time).

The maximum Processor Service Time (Figure 21-12) can be calculated as:

Processor Service Time = Time Slot - AREVT Latency - Setup Time

The Processor Service Time calculation for receive is similar to the calculation for transmit. See Example 21-1 for Processor Service Time calculation using transmit as an example.

Figure 21-12. Processor Service Time Upon Receive DMA Event (AREVT)

A The device uses SYSCLK2 as the McASP system clock source.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.3.8.1.2 Transfers Through the Data Port (DAT)

NOTE: To perform internal transfers through the data port, clear XBUSEL/RBUSEL bit to 0 in the respective XFMT/RFMT registers. Failure to do so will result in software malfunction.

Typically, you will access the McASP XRBUF registers through the data port. To access through the data port, simply have the CPU or DMA access the XRBUF through its data port location. Refer to the device-specific data manual for the exact memory address. Through the data port, the DMA/CPU can service all the serializers through a single address. The McASP automatically cycles through the appropriate serializers.

For transmit operations through the data port, the DMA/CPU should write to the same XBUF data port address to service all of the active transmit serializers. In addition, the DMA/CPU should write to the XBUF for all active transmit serializers in incremental (although not necessarily consecutive) order. For example, if serializers 0, 4, and 5, are set up as active transmitters, the DMA/CPU should write to the XBUF data port address four times with data for serializers 0, 4, and 5, upon each transmit data ready event. This exact servicing order must be followed so that data appears in the appropriate serializers.

Similarly, for receive operations through the data port, the DMA/CPU should read from the same RBUF data port address to service all of the active receive serializers. In addition, reads from the active receive serializers through the data port return data in incremental (although not necessarily consecutive) order. For example, if serializers 1, 2, and 3, are set up as active receivers, the DMA/CPU should read from the RBUF data port address four times to obtain data for serializers 1, 2, and 3, in this exact order, upon each receive data ready event.

When transmitting, the DMA/CPU must write data to each serializer configured as "active" and "transmit" within each time slot. Failure to do so results in a buffer underrun condition (Section 21.3.8.4.2). Similarly, when receiving, data must be read from each serializer configured as "active" and "receive" within each time slot. Failure to do results in a buffer overrun condition (Section 21.3.8.4.3).

To perform internal transfers through the data port, clear XBUSEL/RBUSEL bit to 0 in the respective XFMT/RFMT registers.

21.3.8.1.3 Transfers Through the Configuration Bus (CFG)

NOTE: To perform internal transfers through the configuration bus, set XBUSEL/RBUSEL bit to 1 in the respective XFMT/RFMT registers. Failure to do so will result in software malfunction.

In this method, the DMA/CPU accesses the XRBUF registers through the configuration bus address. The exact XRBUF register address for any particular serializer is determined by adding the offset for that particular serializer to the base address for the particular McASP (found in the device-specific data manual). XRBUF for the serializers configured as transmitters is given the name XBUF*n*. For example, the XRBUF associated with transmit serializer 2 is named XBUF2. Similarly, XRBUF for the serializers configured as receivers is given the name RBUF*n*.

Accessing the XRBUF registers through the data port is different because the CPU/DMA only needs to access one single address. When accessing through the configuration bus, the CPU/DMA must provide the exact XBUF*n* or RBUF*n* address for each access.

When transmitting, DMA/CPU must write data to each serializer configured as "active" and "transmit" within each time slot. Failure to do so results in a buffer underrun condition (Section 21.3.8.4.2). Similarly when receiving, data must be read from each serializer configured as "active" and "receive" within each time slot. Failure to do results in a buffer overrun condition (Section 21.3.8.4.3).

To perform internal transfers through the configuration bus, set XBUSEL/RBUSEL bit to 1 in the respective XFMT/RFMT registers.

21.3.8.1.4 Using the CPU for McASP Servicing

The CPU can be used to service the McASP through interrupt (upon AXINT/ARINT interrupts) or through polling the XDATA bit in the XSTAT register. As discussed in Section 21.3.8.1.2 and Section 21.3.8.1.3, the CPU can access through either the data port or through the configuration bus.

Copyright © 2011, Texas Instruments Incorporated

To use the CPU to service the McASP through interrupts, the XDATA/RDATA bit must be enabled in the respective XINTCTL/RINTCTL registers, to generate interrupts AXINT/ARINT to the CPU upon data ready.

21.3.8.2 McASP Audio FIFO (AFIFO)

The AFIFO contains two FIFOs: one Read FIFO (RFIFO), and one Write FIFO (WFIFO). To ensure backward compatibility with existing software, both the Read and Write FIFOs are disabled by default. See Figure 21-13 for a high-level block diagram of the AFIFO.

The AFIFO may be enabled/disabled and configured via the WFIFOCTL and RFIFOCTL registers. Note that if the Read or Write FIFO is to be enabled, it must be enabled prior to initializing the receive/transmit section of the McASP (see Section 21.3.10.2 for details).

Figure 21-13. McASP Audio FIFO (AFIFO) Block Diagram

21.3.8.2.1 AFIFO Data Transmission

When the Write FIFO is disabled, transmit DMA requests pass through directly from the McASP to the host/DMA controller. Whether the WFIFO is enabled or disabled, the McASP generates transmit DMA requests as needed; the AFIFO is “invisible” to the McASP.

When the Write FIFO is enabled, transmit DMA requests from the McASP are sent to the AFIFO, which in turn generates transmit DMA requests to the host/DMA controller.

If the Write FIFO is enabled, upon a transmit DMA request from the McASP, the WFIFO writes *WNUMDMA* 32-bit words to the McASP if and when there are at least *WNUMDMA* words in the Write FIFO. If there are not, the WFIFO waits until this condition has been satisfied. At that point, it writes *WNUMDMA* words to the McASP. (See description for WFIFOCTL.WNUMDMA in Section 21.4.43.)

If the host CPU writes to the Write FIFO, independent of a transmit DMA request, the WFIFO will accept host writes until full. After this point, excess data will be discarded.

Note that when the WFIFO is first enabled, it will immediately issue a transmit DMA request to the host. This is because it begins in an empty state, and is therefore ready to accept data.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.3.8.2.1.1 Transmit DMA Event Pacer

The AFIFO may be configured to delay making a transmit DMA request to the host until the Write FIFO has enough space for a specified number of words. In this situation, the number of transmit DMA requests to the host or DMA controller is reduced.

If the Write FIFO has space to accept *WNUMEVT* 32-bit words, it generates a transmit DMA request to the host and then waits for a response. Once *WNUMEVT* words have been written to the FIFO, it checks again to see if there is space for *WNUMEVT* 32-bit words. If there is space, it generates another transmit DMA request to the host, and so on. In this fashion, the Write FIFO will attempt to stay filled.

Note that if transmit DMA event pacing is desired, WFIFOCTL.WNUMEVT should be set to a non-zero integer multiple of the value in WFIFOCTL.WNUMDMA. If transmit DMA event pacing is not desired, then the value in WFIFOCTL.WNUMEVT should be set equal to the value in WFIFOCTL.WNUMDMA.

21.3.8.2.2 AFIFO Data Reception

When the Read FIFO is disabled, receive DMA requests pass through directly from McASP to the host/DMA controller. Whether the RFIFO is enabled or disabled, the McASP generates receive DMA requests as needed; the AFIFO is “invisible” to the McASP.

When the Read FIFO is enabled, receive DMA requests from the McASP are sent to the AFIFO, which in turn generates receive DMA requests to the host/DMA controller.

If the Read FIFO is enabled and the McASP makes a receive DMA request, the RFIFO reads *RNUMDMA* 32-bit words from the McASP, if and when the RFIFO has space for *RNUMDMA* words. If it does not, the RFIFO waits until this condition has been satisfied; at that point, it reads *RNUMDMA* words from the McASP. (See description for RFIFOCTL.RNUMDMA in Section 21.4.45.)

If the host CPU reads the Read FIFO, independent of a receive DMA request, and the RFIFO at that time contains less than *RNUMEVT* words, those words will be read correctly, emptying the FIFO.

21.3.8.2.2.1 Receive DMA Event Pacer

The AFIFO may be configured to delay making a receive DMA request to the host until the Read FIFO contains a specified number of words. In this situation, the number of receive DMA requests to the host or DMA controller is reduced.

If the Read FIFO contains at least *RNUMEVT* 32-bit words, it generates a receive DMA request to the host and then waits for a response. Once *RNUMEVT* 32-bit words have been read from the RFIFO, the RFIFO checks again to see if it contains at least another *RNUMEVT* words. If it does, it generates another receive DMA request to the host, and so on. In this fashion, the Read FIFO will attempt to stay empty.

Note that if receive DMA event pacing is desired, RFIFOCTL.RNUMEVT should be set to a non-zero integer multiple of the value in RFIFOCTL.RNUMDMA. If receive DMA event pacing is not desired, then the value in RFIFOCTL.RNUMEVT should be set equal to the value in RFIFOCTL.RNUMDMA.

21.3.8.2.3 Arbitration Between Transmit and Receive DMA Requests

If both the WFIFO and the RFIFO are enabled and a transmit DMA request and receive DMA request occur simultaneously, priority is given to the transmit DMA request. Once a transfer is in progress, it is allowed to complete.

If only the WFIFO is enabled and a transmit DMA request and receive DMA request occur simultaneously, priority is given to the transmit DMA request. Once a transfer is in progress, it is allowed to complete.

If only the RFIFO is enabled and a transmit DMA request and receive DMA request occur simultaneously, priority is given to the receive DMA request. Once a transfer is in progress, it is allowed to complete.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.3.8.3 Formatter

21.3.8.3.1 Transmit Bit Stream Data Alignment

The McASP transmitter supports serial formats of:

- Slot (or Time slot) size = 8, 12, 16, 20, 24, 28, 32 bits.
- Word size ≤ Slot size.
- Alignment: when more bits/slot than bits/words, then:
  - Left aligned = word shifted first, remaining bits are pad.
  - Right aligned = pad bits are shifted first, word occupies the last bits in slot.
- Order: order of bits shifted out:
  - MSB: most-significant bit of word is shifted out first, last bit is LSB.
  - LSB: least-significant bit of word is shifted out last, last bit is MSB.

Hardware support for these serial formats comes from the programmable options in the transmit bitstream format register (XFMT):

- XRVRS: bit reverse (1) or no bit reverse (0).
- XROT: rotate right by 0, 4, 8, 12, 16, 20, 24, or 28 bits.
- XSSZ: transmit slot size of 8, 12, 16, 20, 24, 28, or 32 bits.

XSSZ should always be programmed to match the slot size of the serial stream. The word size is not directly programmed into the McASP, but rather is used to determine the rotation needed in the XROT field.

Table 21-6 and Figure 21-14 show the XRVRS and XROT fields for each serial format and for both integer and Q31 fractional internal representations.

This discussion assumes that all slot size (SLOT in Table 21-6) and word size (WORD in Table 21-6) options are multiples of 4, since the transmit rotate right unit only supports rotation by multiples of 4. However, the bit mask/pad unit does allow for any number of significant digits. For example, a Q31 number may have 19 significant digits (word) and be transmitted in a 24-bit slot; this would be formatted as a word size of 20 bits and a slot size of 24 bits. However, it is possible to set the bit mask unit to only pass the 19 most-significant digits (program the mask value to FFFF E000h). The digits that are not significant can be set to a selected pad value, which can be any one of the significant digits, a fixed value of 0, or a fixed value of 1.

The transmit bit mask/pad unit operates on data as an initial step of the transmit format unit (see Figure 21-9), and the data is aligned in the same representation as it is written to the transmitter by the processor (typically Q31 or integer).

Table 21-6. Transmit Bitstream Data Alignment

Figure 21-14	Bit Stream Order	Bit Stream Alignment	Internal Numeric Representation	XFMT Bit	
				XROT [1]	XRVRS
(a) [2]	MSB first	Left aligned	Q31 fraction	0	1
(b)	MSB first	Right aligned	Q31 fraction	SLOT - WORD	1
(c)	LSB first	Left aligned	Q31 fraction	32 - WORD	0
(d)	LSB first	Right aligned	Q31 fraction	32 - SLOT	0
(e) [2]	MSB first	Left aligned	Integer	WORD	1
(f)	MSB first	Right aligned	Integer	SLOT	1
(g)	LSB first	Left aligned	Integer	0	0
(h)	LSB first	Right aligned	Integer	(32 - (SLOT - WORD)) % 32	0

(1) WORD = Word size rounded up to the nearest multiple of 4; SLOT = slot size; % = modulo operator
(2) To transmit in I2S format, use MSB first, left aligned, and also select XDATDLY = 01 (1 bit delay)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 21-14. Data Flow Through Transmit Format Unit, Illustrated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.3.8.3.2 Receive Bit Stream Data Alignment

The McASP receiver supports serial formats of:

- Slot or time slot size = 8, 12, 16, 20, 24, 28, 32 bits.
- Word size ≤ Slot size.
- Alignment when more bits/slot than bits/words, then:
  - Left aligned = word shifted first, remaining bits are pad.
  - Right aligned = pad bits are shifted first, word occupies the last bits in slot.
- Order of bits shifted out:
  - MSB: most-significant bit of word is shifted out first, last bit is LSB.
  - LSB: least-significant bit of word is shifted out last, last bit is MSB.

Hardware support for these serial formats comes from the programmable options in the receive bitstream format register (RFMT):

- RRVRS: bit reverse (1) or no bit reverse (0).
- RROT: rotate right by 0, 4, 8, 12, 16, 20, 24, or 28 bits.
- RSSZ: receive slot size of 8, 12, 16, 20, 24, 28, or 32 bits.

RSSZ should always be programmed to match the slot size of the serial stream. The word size is not directly programmed into the McASP, but rather is used to determine the rotation needed in the RROT field.

Table 21-7 and Figure 21-15 show the RRVRS and RROT fields for each serial format and for both integer and Q31 fractional internal representations.

This discussion assumes that all slot size and word size options are multiples of 4; since the receive rotate right unit only supports rotation by multiples of 4. However, the bit mask/pad unit does allow for any number of significant digits. For example, a Q31 number may have 19 significant digits (word) and be transmitted in a 24-bit slot; this would be formatted as a word size of 20 bits and a slot size of 24 bits. However, it is possible to set the bit mask unit to only pass the 19 most-significant digits (program the mask value to FFFF E000h). The digits that are not significant can be set to a selected pad value, which can be any one of the significant digits, a fixed value of 0, or a fixed value of 1.

The receive bit mask/pad unit operates on data as the final step of the receive format unit (see Figure 21-8), and the data is aligned in the same representation as it is read from the receiver by the processor (typically Q31 or integer).

Table 21-7. Receive Bitstream Data Alignment

Figure 21-15	Bit Stream Order	Bit Stream Alignment	Internal Numeric Representation	RFMT Bit	
				RROT [1]	RRVRS
(a) [2]	MSB first	Left aligned	Q31 fraction	SLOT	1
(b)	MSB first	Right aligned	Q31 fraction	WORD	1
(c)	LSB first	Left aligned	Q31 fraction	(32 - (SLOT - WORD)) % 32	0
(d)	LSB first	Right aligned	Q31 fraction	0	0
(e) [2]	MSB first	Left aligned	Integer	SLOT - WORD	1
(f)	MSB first	Right aligned	Integer	0	1
(g)	LSB first	Left aligned	Integer	32 - SLOT	0
(h)	LSB first	Right aligned	Integer	32 - WORD	0

(1) WORD = Word size rounded up to the nearest multiple of 4; SLOT = slot size; % = modulo operator
(2) To transmit in I2S format, select MSB first, left aligned, and also select RDATDLY = 01 (1 bit delay)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 21-15. Data Flow Through Receive Format Unit, Illustrated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.3.8.4 Error Handling and Management

To support the design of a robust audio system, the McASP includes error-checking capability for the serial protocol, data underrun, and data overrun. In addition, the McASP includes a timer that continually measures the high-frequency master clock every 32 AHCLKX/AHCLKR clock cycles. The timer value can be read to get a measurement of the clock frequency and has a minimum and maximum range setting that can set an error flag if the master clock goes out of a specified range.

Upon the detection of any one or more errors (software selectable), or the assertion of the AMUTEIN input pin, the AMUTE output pin may be asserted to a high or low level to immediately mute the audio output. In addition, an interrupt may be generated if desired, based on any one or more of the error sources.

21.3.8.4.1 Unexpected Frame Sync Error

An unexpected frame sync occurs when:

- In burst mode, when the next active edge of the frame sync occurs early such that the current slot will not be completed by the time the next slot is scheduled to begin.
- In TDM mode, a further constraint is that the frame sync must occur exactly during the correct bit clock (not a cycle earlier or later) and only before slot 0. An unexpected frame sync occurs if this condition is not met.

When an unexpected frame sync occurs, there are two possible actions depending upon when the unexpected frame sync occurs:

1. Early: An early unexpected frame sync occurs when the McASP is in the process of completing the current frame and a new frame sync is detected (not including overlap that occurs due to a 1 or 2 bit frame sync delay). When an early unexpected frame sync occurs:
   - Error interrupt flag is set (XSYNCERR, if an unexpected transmit frame sync occurs; RSYNCERR, if an unexpected receive frame sync occurs).
   - Current frame is not resynchronized. The number of bits in the current frame is completed. The next frame sync, which occurs after the current frame is completed, will be resynchronized.
2. Late: A late unexpected frame sync occurs when there is a gap or delay between the last bit of the previous frame and the first bit of the next frame. When a late unexpected frame sync occurs (as soon as the gap is detected):
   - Error interrupt flag is set (XSYNCERR, if an unexpected transmit frame sync occurs; RSYNCERR, if an unexpected receive frame sync occurs).
   - Resynchronization occurs upon the arrival of the next frame sync.

Late frame sync is detected the same way in both burst mode and TDM mode; however, in burst mode, late frame sync is not meaningful and its interrupt enable should not be set.

21.3.8.4.2 Buffer Underrun Error - Transmitter

A buffer underrun can only occur for serializers programmed to be transmitters. A buffer underrun occurs when the serializer is instructed by the transmit state machine to transfer data from XRBUF[n] to XRSR[n], but XRBUF[n] has not yet been written with new data since the last time the transfer occurred. When this occurs, the transmit state machine sets the XUNDRN flag.

An underrun is checked only once per time slot. The XUNDRN flag is set when an underrun condition occurs. Once set, the XUNDRN flag remains set until the processor explicitly writes a 1 to the XUNDRN bit to clear the XUNDRN bit.

In DIT mode, a pair of BMC zeros is shifted out when an underrun occurs (four bit times at 128 × fs). By shifting out a pair of zeros, a clock may be recovered on the receiver. To recover, reset the McASP and start again with the proper initialization.

In TDM mode, during an underrun case, a long stream of zeros are shifted out causing the DACs to mute. To recover, reset the McASP and start again with the proper initialization.

Copyright © 2011, Texas Instruments Incorporated

21.3.8.4.3 Buffer Overrun Error - Receiver

A buffer overrun can only occur for serializers programmed to be receivers. A buffer overrun occurs when the serializer is instructed to transfer data from XRSR[n] to XRBUF[n], but XRBUF[n] has not yet been read by either the DMA or the processor. When this occurs, the receiver state machine sets the ROVRN flag. However, the individual serializer writes over the data in the XRBUF[n] register (destroying the previous sample) and continues shifting.

An overrun is checked only once per time slot. The ROVRN flag is set when an overrun condition occurs. It is possible that an overrun occurs on one time slot but then the processor catches up and does not cause an overrun on the following time slots. However, once the ROVRN flag is set, it remains set until the processor explicitly writes a 1 to the ROVRN bit to clear the ROVRN bit.

21.3.8.4.4 DMA Error - Transmitter

A transmit DMA error, as indicated by the XDMAERR flag in the XSTAT register, occurs when the DMA (or CPU) writes more words to the DAT port of the McASP than it should. For each DMA event, the DMA should write exactly as many words as there are serializers enabled as transmitters.

XDMAERR indicates that the DMA (or CPU) wrote too many words to the McASP for a given transmit DMA event. Writing too few words results in a transmit underrun error setting XUNDRN in XSTAT.

While XDMAERR occurs infrequently, an occurrence indicates a serious loss of synchronization between the McASP and the DMA or CPU. You should reinitialize both the McASP transmitter and the DMA to resynchronize them.

21.3.8.4.5 DMA Error - Receiver

A receive DMA error, as indicated by the RDMAERR flag in the RSTAT register, occurs when the DMA (or CPU) reads more words from the DAT port of the McASP than it should. For each DMA event, the DMA should read exactly as many words as there are serializers enabled as receivers.

RDMAERR indicates that the DMA (or CPU) read too many words from the McASP for a given receive DMA event. Reading too few words results in a receiver overrun error setting ROVRN in RSTAT.

While RDMAERR occurs infrequently, an occurrence indicates a serious loss of synchronization between the McASP and the DMA or CPU. You should reinitialize both the McASP receiver and the DMA to resynchronize them.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.3.8.4.6 Clock Failure Detection

21.3.8.4.6.1 Clock-Failure Check Startup

It is expected, initially, that the clock-failure circuits will generate an error until at least one measurement has been taken. Therefore, the clock failure interrupts, clock switch, and mute functions should not immediately be enabled, but be enabled only after a specific startup procedure. The startup procedure is:

1. For the transmit clock failure check:
   (a) Configure transmit clock failure detect logic (XMIN, XMAX, XPS) in the transmit clock check control register (XCLKCHK).
   (b) Clear transmit clock failure flag (XCKFAIL) in the transmit status register (XSTAT).
   (c) Wait until first measurement is taken (> 32 AHCLKX clock periods).
   (d) Verify no clock failure is detected.
   (e) Repeat steps b–d until clock is running and is no longer issuing clock failure errors.
   (f) After the transmit clock is measured and falls within the acceptable range, the following may be enabled:
      (i) transmit clock failure interrupt enable bit (XCKFAIL) in the transmitter interrupt control register (XINTCTL).
      (ii) transmit clock failure detect autoswitch enable bit (XCKFAILSW) in the transmit clock check control register (XCLKCHK).
      (iii) mute option (XCKFAIL) in the mute control register (AMUTE).
2. For the receive clock failure check:
   (a) Configure receive clock failure detect logic (RMIN, RMAX, RPS) in the receive clock check control register (RCLKCHK).
   (b) Clear receive clock failure flag (RCKFAIL) in the receive status register (RSTAT).
   (c) Wait until first measurement is taken (> 32 AHCLKR clock periods).
   (d) Verify no clock failure is detected.
   (e) Repeat steps b–d until clock is running and is no longer issuing clock failure errors.
   (f) After the receive clock is measured and falls within the acceptable range, the following may be enabled:
      (i) receive clock failure interrupt enable bit (RCKFAIL) in the receiver interrupt control register (RINTCTL).
      (ii) mute option (RCKFAIL) in the mute control register (AMUTE).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.3.8.4.6.2 Transmit Clock Failure Check and Recovery

The transmit clock failure check circuit (Figure 21-16) works off both the internal McASP system clock and the external high-frequency serial clock (AHCLKX). It continually counts the number of system clocks for every 32 high rate serial clock (AHCLKX) periods, and stores the count in XCNT of the transmit clock check control register (XCLKCHK) every 32 high rate serial clock cycles.

The logic compares the count against a user-defined minimum allowable boundary (XMIN), and automatically flags an interrupt (XCKFAIL in XSTST) when an out-of-range condition occurs. An out-of-range minimum condition occurs when the count is smaller than XMIN. The logic continually compares the current count (from the running system clock counter) against the maximum allowable boundary (XMAX). This is in case the external clock completely stops, so that the counter value is not copied to XCNT. An out-of-range maximum condition occurs when the count is greater than XMAX. Note that the XMIN and XMAX fields are 8-bit unsigned values, and the comparison is performed using unsigned arithmetic.

An out-of-range count may indicate either that an unstable clock was detected, or that the audio source has changed and a new sample rate is being used.

In order for the transmit clock failure check circuit to operate correctly, the high-frequency serial clock divider must be taken out of reset regardless if AHCLKX is internally generated or externally sourced.

Figure 21-16. Transmit Clock Failure Detection Circuit Block Diagram

A Refer to device data manual for the McASP system clock source. This is not the same as AUXCLK.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

The following actions are taken if a clock failure is detected:

1. Transmit clock failure flag (XCKFAIL) in XSTAT is set. This causes an interrupt if transmit clock failure interrupt enable bit (XCKFAIL) in XINTCTL is set.

In addition (only supported for DIT mode), if the transmit clock failure detect autoswitch enable bit (XCKFAILSW) in XCLKCHK is set, the following additional steps are taken to change the clock source from external to internal:

1. High-frequency transmit clock source bit (HCLKXM) in AHCLKXCTL is set to 1 and internal serial clock divider is selected. However, AHCLKX pin direction does not change to an output while XCKFAIL is set.
2. The internal clock divider is reset, so that the next clock it produces is a full period. However, the transmit clock divide ratio bits (HCLKXDIV) in AHCLKXCTL are not affected, so the internal clock divider generates clocks at the rate configured.
3. The transmit section is reset for a single serial clock period.
4. The transmit section is released from reset and attempts to begin transmitting. If data is available, it begins transmitting immediately; otherwise, it enters the underrun state. An initial underrun is certain to occur, the pattern 1100 (BMC zeroes) should be shifted out initially.

To change back to an external clock, take the following actions:

1. Wait for the external clock to stabilize again. This can be checked by polling the transmit clock count (XCNT) in XCLKCHK.
2. Reset the transmit section according to the startup procedure in Section 21.3.8.4.6.1.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.3.8.4.6.3 Receive Clock Failure Check and Recovery

The receive clock failure check circuit (Figure 21-17) works off both the internal McASP system clock and the external high-frequency serial clock (AHCLKR). It continually counts the number of system clocks for every 32 high rate serial clock (AHCLKR) periods, and stores the count in RCNT of the receive clock check control register (RCLKCHK) every 32 high rate serial clock cycles.

The logic compares the count against a user-defined minimum allowable boundary (RMIN) and automatically flags an interrupt (RCKFAIL in RSTAT) when an out-of-range condition occurs. An out-of-range minimum condition occurs when the count is smaller than RMIN. The logic continually compares the current count (from the running system clock counter) against the maximum allowable boundary (RMAX). This is in case the external clock completely stops, so that the counter value is not copied to RCNT. An out-of-range maximum condition occurs when the count is greater than RMAX. Note that the RMIN and RMAX fields are 8-bit unsigned values, and the comparison is performed using unsigned arithmetic.

An out-of-range count may indicate either that an unstable clock was detected or that the audio source has changed and a new sample rate is being used.

In order for the receive clock failure check circuit to operate correctly, the high-frequency serial clock divider must be taken out of reset regardless if AHCLKR is internally generated or externally sourced.

Figure 21-17. Receive Clock Failure Detection Circuit Block Diagram

A Refer to device data manual for the McASP system clock source. This is not the same as AUXCLK.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.3.8.5 Loopback Modes

The McASP features a digital loopback mode (DLB) that allows testing of the McASP code in TDM mode with a single processor device. In loopback mode, output of the transmit serializers is connected internally to the input of the receive serializers. Therefore, you can check the receive data against the transmit data to ensure that the McASP settings are correct. Digital loopback mode applies to TDM mode only (2 to 32 slots in a frame). It does not apply to DIT mode (XMOD = 180h) or burst mode (XMOD = 0).

Figure 21-18 shows the basic logical connection of the serializers in loopback mode. Two types of loopback connections are possible, selected by the ORD bit in the digital loopback control register (DLBCTL) as follows:

- ORD = 0: Outputs of odd serializers are connected to inputs of even serializers. If this mode is selected, you should configure odd serializers to be transmitters and even serializers to be receivers.
- ORD = 1: Outputs of even serializers are connected to inputs of odd serializers. If this mode is selected, you should configure even serializers to be transmitters and odd serializers to be receivers.

Data can be externally visible at the I/O pin of the transmit serializer if the pin is configured as a McASP output pin by setting the corresponding PFUNC bit to 0 and PDIR bit to 1.

In loopback mode, the transmit clock and frame sync are used by both the transmit and receive sections of the McASP. The transmit and receive sections operate synchronously. This is achieved by setting the MODE bit of the DLBCTL register to 01b and the ASYNC bit of the ACLKXCTL register to 0.

Figure 21-18. Serializers in Loopback Mode

(a) DLBEN = 1 (loopback enabled) and ORD = 0 (even receive, odd transmit)

(b) DLBEN = 1 (loopback enabled) and ORD = 1 (odd receive, even transmit)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.3.8.5.1 Loopback Mode Configurations

This is a summary of the settings required for digital loopback mode for TDM format:

- The DLBEN bit in DLBCTL must be set to 1 to enable loopback mode.
- The MODE bits in DLBCTL must be set to 01b for both the transmit and receive sections to use the transmit clock and frame sync generator.
- The ORD bit in DLBCTL must be programmed appropriately to select odd or even serializers to be transmitters or receivers. The corresponding serializers must be configured accordingly.
- The ASYNC bit in ACLKXCTL must be cleared to 0 to ensure synchronous transmit and receive operations.
- RMOD field in AFSRCTL and XMOD field in AFSXCTL must be set to 2h to 20h to indicate TDM mode. Loopback mode does not apply to DIT or burst mode.

21.3.9 Reset Considerations

The McASP has two reset sources: software reset and hardware reset.

21.3.9.1 Software Reset Considerations

The transmitter and receiver portions of the McASP may be put in reset through the global control register (GBLCTL). Note that a valid serial clock must be supplied to the desired portion of the McASP (transmit and/or receive) in order to assert the software reset bits in GBLCTL. see Section 21.3.10.2for details on how to ensure reset has occurred.

The entire McASP module may also be reset through the Power and Sleep Controller (PSC). Note that from the McASP perspective, this reset appears as a hardware reset to the entire module.

21.3.9.2 Hardware Reset Considerations

When the McASP is reset due to device reset, the entire serial port (including the transmitter and receiver state machines, and other registers) is reset.

21.3.10 Setup and Initialization

This section discusses steps necessary to use the McASP module.

21.3.10.1 Considerations When Using a McASP

The following is a list of things to be considered for systems using a McASP:

21.3.10.1.1 Clocks

For each receive and transmit section:

- External or internal generated bit clock and high frequency clock?
- If internally generated, what is the bit clock speed and the high frequency clock speed?
- Clock polarity?
- External or internal generated frame sync?
- If internally generated, what is frame sync speed?
- Frame sync polarity?
- Frame sync width?
- Transmit and receive sync or asynchronous?

21.3.10.1.2 Data Pins

For each pin of each McASP:

- McASP or GPIO?
- Input or output?

Copyright © 2011, Texas Instruments Incorporated

21.3.10.1.3 Data Format

For each transmit and receive data:

- Internal numeric representation (integer, Q31 fraction)?
- I2S or DIT (transmit only)?
- Time slot delay (0, 1, or 2 bit)?
- Alignment (left or right)?
- Order (MSB first, LSB first)?
- Pad (if yes, pad with what value)?
- Slot size?
- Rotate?
- Mask?

21.3.10.1.4 Data Transfers

- Internal: DMA or CPU?
- External: TDM or burst?
- Bus: configuration bus (CFG) or data port (DAT)?

21.3.10.2 Transmit/Receive Section Initialization

You must follow the following steps to properly configure the McASP. If external clocks are used, they should be present prior to the following initialization steps.

1. Reset McASP to default values by setting GBLCTL = 0.
2. Configure all McASP registers except GBLCTL in the following order:
   (a) Power down and emulation management: PWRDEMU.
   (b) Receive registers: RMASK, RFMT, AFSRCTL, ACLKRCTL, AHCLKRCTL, RTDM, RINTCTL, RCLKCHK. If external clocks AHCLKR and/or ACLKR are used, they must be running already for proper synchronization of the GBLCTL register.
   (c) Transmit registers: XMASK, XFMT, AFSXCTL, ACLKXCTL, AHCLKXCTL, XTDM, XINTCTL, XCLKCHK. If external clocks AHCLKX and/or ACLKX are used, they must be running already for proper synchronization of the GBLCTL register.
   (d) Serializer registers: SRCTL[n].
   (e) Global registers: Registers PFUNC, PDIR, DITCTL, DLBCTL, AMUTE. Note that PDIR should only be programmed after the clocks and frames are set up in the steps above. This is because the moment a clock pin is configured as an output in PDIR, the clock pin starts toggling at the rate defined in the corresponding clock control register. Therefore you must ensure that the clock control register is configured appropriately before you set the pin to be an output. A similar argument applies to the frame sync pins. Also note that the reset state for the transmit high-frequency clock divide register (HCLKXDIV) is divide-by-1, and the divide-by-1 clocks are not gated by the transmit high-frequency clock divider reset enable (XHCLKRST).
   (f) DIT registers: For DIT mode operation, set up registers DITCSRA[n], DITCSRB[n], DITUDRA[n], and DITUDRB[n].
3. Start the respective high-frequency serial clocks AHCLKX and/or AHCLKR. This step is necessary even if external high-frequency serial clocks are used:
   (a) Take the respective internal high-frequency serial clock divider(s) out of reset by setting the RHCLKRST bit for the receiver and/or the XHCLKRST bit for the transmitter in GBLCTL. All other bits in GBLCTL should be held at 0.
   (b) Read back from GBLCTL to ensure the bit(s) to which you wrote are successfully latched in GBLCTL before you proceed.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

4. Start the respective serial clocks ACLKX and/or ACLKR. This step can be skipped if external serial clocks are used and they are running:
   (a) Take the respective internal serial clock divider(s) out of reset by setting the RCLKRST bit for the receiver and/or the XCLKRST bit for the transmitter in GBLCTL. All other bits in GBLCTL should be left at the previous state.
   (b) Read back from GBLCTL to ensure the bit(s) to which you wrote are successfully latched in GBLCTL before you proceed.
5. Setup data acquisition as required:
   (a) If DMA is used to service the McASP, set up data acquisition as desired and start the DMA in this step, before the McASP is taken out of reset.
   (b) If CPU interrupt is used to service the McASP, enable the transmit and/ or receive interrupt as required.
   (c) If CPU polling is used to service the McASP, no action is required in this step.
6. Activate serializers.
   (a) Before starting, clear the respective transmitter and receiver status registers by writing XSTAT = FFFFh and RSTAT = FFFFh.
   (b) Take the respective serializers out of reset by setting the RSRCLR bit for the receiver and/or the XSRCLR bit for the transmitter in GBLCTL. All other bits in GBLCTL should be left at the previous state.
   (c) Read back from GBLCTL to ensure the bit(s) to which you wrote are successfully latched in GBLCTL before you proceed.
7. Verify that all transmit buffers are serviced. Skip this step if the transmitter is not used. Also, skip this step if time slot 0 is selected as inactive (special cases, see Figure 21-6, second waveform). As soon as the transmit serializer is taken out of reset, XDATA in the XSTAT register is set, indicating that XBUF is empty and ready to be serviced. The XDATA status causes an DMA event AXEVT to be generated, and can cause an interrupt AXINT to be generated if it is enabled in the XINTCTL register.
   (a) If DMA is used to service the McASP, the DMA automatically services the McASP upon receiving AXEVT. Before proceeding in this step, you should verify that the XDATA bit in the XSTAT is cleared to 0, indicating that all transmit buffers are already serviced by the DMA.
   (b) If CPU interrupt is used to service the McASP, interrupt service routine is entered upon the AXINT interrupt. The interrupt service routine should service the XBUF registers. Before proceeding in this step, you should verify that the XDATA bit in XSTAT is cleared to 0, indicating that all transmit buffers are already serviced by the CPU.
   (c) If CPU polling is used to service the McASP, the XBUF registers should be written to in this step.
8. Release state machines from reset.
   (a) Take the respective state machine(s) out of reset by setting the RSMRST bit for the receiver and/or the XSMRST bit for the transmitter in GBLCTL. All other bits in GBLCTL should be left at the previous state.
   (b) Read back from GBLCTL to ensure the bit(s) to which you wrote are successfully latched in GBLCTL before you proceed.
9. Release frame sync generators from reset. Note that it is necessary to release the internal frame sync generators from reset, even if an external frame sync is being used, because the frame sync error detection logic is built into the frame sync generator.
   (a) Take the respective frame sync generator(s) out of reset by setting the RFRST bit for the receiver, and/or the XFRST bit for the transmitter in GBLCTL. All other bits in GBLCTL should be left at the previous state.
   (b) Read back from GBLCTL to ensure the bit(s) to which you wrote are successfully latched in GBLCTL before you proceed.
10. Upon the first frame sync signal, McASP transfers begin. The McASP synchronizes to an edge on the frame sync pin, not the level on the frame sync pin. This makes it easy to release the state machine and frame sync generators from reset.
   (a) For example, if you configure the McASP for a rising edge transmit frame sync, then you do not need to wait for a low level on the frame sync pin before releasing the McASP transmitter state

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

machine and frame sync generators from reset.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.3.10.3 Separate Transmit and Receive Initialization

In many cases, it is desirable to separately initialize the McASP transmitter and receiver. For example, you may delay the initialization of the transmitter until the type of data coming in on the receiver is recognized. Or a change in the incoming data stream on the receiver may necessitate a reinitialization of the transmitter.

In this case, you may still follow the sequence outlined in Section 21.3.10.2, but use it for each section (transmit, receive) individually. The GBLCTL register is aliased to RGBLCTL and XGBLCTL to facilitate separate initialization of transmit and receive sections.

21.3.10.4 Importance of Reading Back GBLCTL

In Section 21.3.10.2, steps 3b, 4b, 6c, 8b, and 9b state that GBLCTL should be read back until the bits that were written are successfully latched. This is important, because the transmitter and receiver state machines run off of the respective bit clocks, which are typically about tens to hundreds of times slower than the processor's internal bus clock. Therefore, it takes many cycles between when the processor writes to GBLCTL (or RGBLCTL and XGBLCTL), and when the McASP actually recognizes the write operation. If you skip this step, then the McASP may never see the reset bits in the global control registers get asserted and de-asserted; resulting in an uninitialized McASP.

Therefore, the logic in McASP has been implemented such that once the processor writes GBLCTL, RGBLCTL, or XGBLCTL, the resulting write is not visible by reading back GBLCTL until the McASP has recognized the change. This typically requires two bit clocks plus two processor bus clocks to occur.

Also, if the bit clocks can be completely stopped, any software that polls GBLCTL should be implemented with a time-out. If GBLCTL does not have a time-out, and the bit clock stops, the changes written to GBLCTL will not be reflected until the bit clock restarts.

Finally, please note that while RGBLCTL and XGBLCTL allow separate changing of the receive and transmit halves of GBLCTL, they also immediately reflect the updated value (useful for debug purposes). Only GBLCTL can be used for the read back step.

21.3.10.5 Synchronous Transmit and Receive Operation (ASYNC = 0)

When ASYNC = 0 in ACLKXCTL, the transmit and receive sections operate synchronously from the transmit section clock and transmit frame sync signals (Figure 21-2). The receive section may have a different (but compatible in terms of slot size) data format.

When ASYNC = 0, the receive frame sync generator is internally disabled. If the AFSX pin is configured as an output, it serves as the frame sync signal for both transmit and receive. The AFSR pin should not be used because the transmit frame sync generator output, which is used by both the transmitter and the receiver when ASYNC = 0, is not propagated to the AFSR pin (Figure 21-4).

When ASYNC = 0, the transmit and receive sections must share some common settings, since they both use the same clock and frame sync signals:

- DITEN = 0 in DITCTL (TDM mode is enabled).
- The total number of bits per frame must be the same (that is, RSSZ × RMOD must equal XSSZ × XMOD).
- Both transmit and receive should either be specified as burst or TDM mode, but not mixed.
- The settings in ACLKRCTL are irrelevant.
- FSXM must match FSRM.
- FXWID must match FRWID.

For all other settings, the transmit and receive sections may be programmed independently.

21.3.10.6 Asynchronous Transmit and Receive Operation (ASYNC = 1)

When ASYNC = 1 in ACLKXCTL, the transmit and receive sections operate completely independently and have separate clock and frame sync signals (Figure 21-2, Figure 21-3, and Figure 21-4). The events generated by each section come asynchronously.

Copyright © 2011, Texas Instruments Incorporated

21.3.11 Interrupts

21.3.11.1 Interrupt Multiplexing

The processor includes an interrupt controller (INTC) to manage CPU interrupts. The INTC maps the device events to 12 CPU interrupts. The McASP generates 4 interrupts to the processor.

The event inputs can be routed to 12 maskable interrupts to the CPU (INT[15:4]). The INTC interrupt selector allows the McASP system events to be routed to any of the 12 CPU interrupt inputs. Furthermore, the INTC provides status flags and allows the combination of events, as shown in Figure 21-19. You must properly configure the INTC by enabling, masking, and routing the McASP system events to the desired CPU interrupts.

Figure 21-19. Interrupt Multiplexing

21.3.11.2 Transmit Data Ready Interrupt

The transmit data ready interrupt (XDATA) is generated if XDATA is 1 in the XSTAT register and XDATA is also enabled in XINTCTL. Section 21.3.8.1.1 provides details on when XDATA is set in the XSTAT register.

A transmit start of frame interrupt (XSTAFRM) is triggered by the recognition of transmit frame sync. A transmit last slot interrupt (XLAST) is a qualified version of the data ready interrupt (XDATA). It has the same behavior as the data ready interrupt, but is further qualified by having the data requested belonging to the last slot (the slot that just ended was next-to-last TDM slot, current slot is last slot).

21.3.11.3 Receive Data Ready Interrupt

The receive data ready interrupt (RDATA) is generated if RDATA is 1 in the RSTAT register and RDATA is also enabled in RINTCTL. Section 21.3.8.1.2 provides details on when RDATA is set in the RSTAT register.

A receiver start of frame interrupt (RSTAFRM) is triggered by the recognition of a receiver frame sync. A receiver last slot interrupt (RLAST) is a qualified version of the data ready interrupt (RDATA). It has the same behavior as the data ready interrupt, but is further qualified by having the data in the buffer come from the last TDM time slot (the slot that just ended was last TDM slot).

21.3.11.4 Error Interrupts

Upon detection, the following error conditions generate interrupt flags:

- In the receive status register (RSTAT):
  - Receiver overrun (ROVRN).
  - Unexpected receive frame sync (RSYNCERR).
  - Receive clock failure (RCKFAIL).
  - Receive DMA error (RDMAERR).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

- In the transmit status register (XSTAT):
  - Transmit underrun (XUNDRN).
  - Unexpected transmit frame sync (XSYNCERR).
  - Transmit clock failure (XCKFAIL).
  - Transmit DMA error (XDMAERR).

Each interrupt source also has a corresponding enable bit in the receive interrupt control register (RINTCTL) and transmit interrupt control register (XINTCTL). If the enable bit is set in RINTCTL or XINTCTL, an interrupt is requested when the interrupt flag is set in RSTAT or XSTAT. If the enable bit is not set, no interrupt request is generated. However, the interrupt flag may be polled.

21.3.11.5 Audio Mute (AMUTE) Function

The McASP includes an automatic audio mute function (Figure 21-20) that asserts in hardware the AMUTE pin to a preprogrammed output state, as selected by the MUTEN bit in the audio mute control register (AMUTE). The AMUTE pin is asserted when one of the interrupt flags is set or an external device issues an error signal on the AMUTEIN input. Typically, the AMUTEIN input is shared with a device interrupt pin (for example EXT_INT4).

The AMUTEIN input allows the on-chip logic to consider a mute input from other devices in the system, so that all errors may be considered. The AMUTEIN input has a programmable polarity to allow it to adapt to different devices, as selected by the INPOL bit in AMUTE, and it must be enabled explicitly.

In addition to the external AMUTEIN input, the AMUTE pin output may be asserted when one of the error interrupt flags is set and its mute function is enabled in AMUTE.

When one or more of the errors is detected and enabled, the AMUTE pin is driven to an active state that is selected by MUTEN in AMUTE. The active polarity of the AMUTE pin is programmable by MUTEN (and the inactive polarity is the opposite of the active polarity). The AMUTE pin remains driven active until software clears all the error interrupt flags that are enabled to mute, and until the AMUTEIN is inactive.

Figure 21-20. Audio Mute (AMUTE) Block Diagram

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.3.11.6 Multiple Interrupts

This only applies to interrupts and not to DMA requests. The following terms are defined:

- **Active Interrupt Request:** a flag in RSTAT or XSTAT is set and the interrupt is enabled in RINTCTL or XINTCTL.
- **Outstanding Interrupt Request:** An interrupt request has been issued on one of the McASP transmit/receive interrupt ports, but that request has not yet been serviced.
- **Serviced:** The CPU writes to RSTAT or XSTAT to clear one or more of the active interrupt request flags.

The first interrupt request to become active for the transmitter with the interrupt flag set in XSTAT and the interrupt enabled in XINTCTL generates a request on the McASP transmit interrupt port AXINT.

If more than one interrupt request becomes active in the same cycle, a single interrupt request is generated on the McASP transmit interrupt port. Subsequent interrupt requests that become active while the first interrupt request is outstanding do not immediately generate a new request pulse on the McASP transmit interrupt port.

The transmit interrupt is serviced with the CPU writing to XSTAT. If any interrupt requests are active after the write, a new request is generated on the McASP transmit interrupt port.

The receiver operates in a similar way, but using RSTAT, RINTCTL, and the McASP receive interrupt port ARINT.

One outstanding interrupt request is allowed on each port, so a transmit and a receive interrupt request may both be outstanding at the same time.

21.3.12 EDMA Event Support

21.3.12.1 EDMA Events

There are 6 EDMA events.

21.3.12.2 Using the DMA for McASP Servicing

The most typical scenario is to use the DMA to service the McASP through the data port, although the DMA can also service the McASP through the configuration bus. Two possibilities exist for using the DMA events to service the McASP:

1. Use **AXEVT/AREVT**: Triggered upon each XDATA/RDATA transition from 0 to 1.
2. Use **AXEVTO/AREVTO** and **AXEVTE/AREVTE**: Alternating AXEVT/AREVT events for odd/even slots. Upon AXEVT/AREVT, AXEVTO/AREVTO is triggered if the event is for an odd channel, and AXEVTE/AREVTE is triggered if the event is for an even channel.

NOTE: Check the device-specific data manual to see if AXEVTO/AREVTO and AXEVTE/AREVTE are supported. These are optional.

Figure 21-21 and Figure 21-22 show an example audio system with six audio channels (LF, RF, LS, RS, C, and LFE) transmitted from three AXRn pins on the McASP. Figure 21-21 and Figure 21-22 show when events AXEVT, AXEVTO, and AXEVTE are triggered. Figure 21-21 and Figure 21-22 also apply for the receive audio channels and show when events AREVT, AREVTO, and AREVTE are triggered.

You can either use the DMA to service the McASP upon events AXEVT and AREVT (Figure 21-21) or upon events AXEVTO, AREVTO, AXEVTE, and AREVTE (Figure 21-22).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 21-21. DMA Events in an Audio Example–Two Events (Scenario 1)

Figure 21-22. DMA Events in an Audio Example–Four Events (Scenario 2)

In scenario 1 (Figure 21-21), a DMA event AXEVT/AREVT is triggered on each time slot. In the example, AXEVT is triggered for each of the transmit audio channel time slot (Time slot for channels LF, LS, and C; and time slot for channels RF, RS, LFE). Similarly, AREVT is triggered for each of the receive audio channel time slot. Scenario 1 allows for the use of a single DMA to transmit all audio channels, and a single DMA to receive all audio channels.

In scenario 2 (Figure 21-22), two alternating DMA events are triggered for each time slot. In the example, AXEVTE (even) is triggered for the time slot for the even audio channels (LF, LS, C) and AXEVTO (odd) is triggered for the time slot for the odd audio channels (RF, RS, LFE). AXEVTO and AXEVTE alternate in time. The same is true in the receive direction with the use of AREVTO and AREVTE. This scenario allows for the use of two DMA channels (odd and even) to transmit all audio channels, and two DMA channels to receive all audio channels.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Here are some guidelines on using the different DMA events:

- Either use AXEVT, or the combination of AXEVTO and AXEVTE, to service the McASP. Never use all three at the same time. Similarly for receive, either use AREVT, or the combination of AREVTO and AREVTE.
- The McASP generates transmit DMA events independently from receive DMA events; therefore, separate schemes can be used for transmit and receive DMA. For example, scenario 1 could be used for the transmit data (AXEVT) and scenario 2 could be used for the receive data (AREVTO, AREVTE), and conversely.

Note the difference between DMA event generation and the CPU interrupt generation. DMA events are generated automatically upon data ready; whereas CPU interrupt generation needs to be enabled in the XINTCTL/RINTCTL register.

In Figure 21-22, scenario 2, each transmit DMA request is for data in the next time slot, while each receive DMA request is for data in the previous time slot. For example, Figure 21-23 shows a circled AXEVTE event for an even time slot transmit DMA request. The transmitter always requests a DMA transfer for data it will need to transmit during the next time slot. So, in this example, the circled event AXEVTE is a request for data for samples LF2, LS2, and C2.

On the other hand, the circled AREVTE event is an even time slot receive DMA request. The receiver always requests a DMA transfer for data it received during the previous time slot. In this example, the circled event AREVTE is a request for samples LF1, LS1, and C1.

Figure 21-23. DMA Events in an Audio Example

21.3.13 Power Management

The McASP can be placed in reduced power modes to conserve power during periods of low activity.

21.3.14 Emulation Considerations

NOTE: The receive buffer registers (RBUF*n*) and transmit buffer registers (XBUF*n*) should not be accessed by the emulator when the McASP is running. Such an access will cause the RRDY/XRDY bit in the serializer control register *n* (SRCTL*n*) to be updated.

The McASP does not support the emulation suspend.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4 McASP Registers

Control registers for the McASP are summarized in Table 21-8. The control registers are accessed through the configuration bus of the device. The receive buffer registers (RBUF*n*) and transmit buffer registers (XBUF*n*) can also be accessed through the data port of the device, as listed in Table 21-9. See the device-specific data manual for the memory address of these registers.

Control registers for the McASP Audio FIFO (AFIFO) are summarized in Table 21-10. Note that the AFIFO Write FIFO (WFIFO) and Read FIFO (RFIFO) have independent control and status registers. The AFIFO control registers are accessed through the peripheral configuration port. See your device-specific data manual for the memory address of these registers.

Table 21-8. McASP Registers Accessed Through Configuration Bus

Offset	Acronym	Register Description	Section
0h	REV	Revision identification register	Section 21.4.1
10h	PFUNC	Pin function register	Section 21.4.2
14h	PDIR	Pin direction register	Section 21.4.3
18h	PDOUT	Pin data output register	Section 21.4.4
1Ch	PDIN	Read returns: Pin data input register	Section 21.4.5
1Ch	PDSET	Writes affect: Pin data set register (alternate write address: PDOUT)	Section 21.4.6
20h	PDCLR	Pin data clear register (alternate write address: PDOUT)	Section 21.4.7
44h	GBLCTL	Global control register	Section 21.4.8
48h	AMUTE	Audio mute control register	Section 21.4.9
4Ch	DLBCTL	Digital loopback control register	Section 21.4.10
50h	DITCTL	DIT mode control register	Section 21.4.11
60h	RGBLCTL	Receiver global control register: Alias of GBLCTL, only receive bits are affected - allows receiver to be reset independently from transmitter	Section 21.4.12
64h	RMASK	Receive format unit bit mask register	Section 21.4.13
68h	RFMT	Receive bit stream format register	Section 21.4.14
6Ch	AFSRCTL	Receive frame sync control register	Section 21.4.15
70h	ACLKRCTL	Receive clock control register	Section 21.4.16
74h	AHCLKRCTL	Receive high-frequency clock control register	Section 21.4.17
78h	RTDM	Receive TDM time slot 0-31 register	Section 21.4.18
7Ch	RINTCTL	Receiver interrupt control register	Section 21.4.19
80h	RSTAT	Receiver status register	Section 21.4.20
84h	RSLOT	Current receive TDM time slot register	Section 21.4.21
88h	RCLKCHK	Receive clock check control register	Section 21.4.22
8Ch	REVTCTL	Receiver DMA event control register	Section 21.4.23
A0h	XGBLCTL	Transmitter global control register. Alias of GBLCTL, only transmit bits are affected - allows transmitter to be reset independently from receiver	Section 21.4.24
A4h	XMASK	Transmit format unit bit mask register	Section 21.4.25
A8h	XFMT	Transmit bit stream format register	Section 21.4.26

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 21-8. McASP Registers Accessed Through Configuration Bus (continued)

Offset	Acronym	Register Description	Section
ACh	AFSXCTL	Transmit frame sync control register	Section 21.4.27
B0h	ACLKXCTL	Transmit clock control register	Section 21.4.28
B4h	AHCLKXCTL	Transmit high-frequency clock control register	Section 21.4.29
B8h	XTDM	Transmit TDM time slot 0-31 register	Section 21.4.30
BCh	XINTCTL	Transmitter interrupt control register	Section 21.4.31
C0h	XSTAT	Transmitter status register	Section 21.4.32
C4h	XSLOT	Current transmit TDM time slot register	Section 21.4.33
C8h	XCLKCHK	Transmit clock check control register	Section 21.4.34
CCh	XEVTCTL	Transmitter DMA event control register	Section 21.4.35
100h	DITCSRA0	Left (even TDM time slot) channel status register (DIT mode) 0	Section 21.4.37
104h	DITCSRA1	Left (even TDM time slot) channel status register (DIT mode) 1	Section 21.4.37
108h	DITCSRA2	Left (even TDM time slot) channel status register (DIT mode) 2	Section 21.4.37
10Ch	DITCSRA3	Left (even TDM time slot) channel status register (DIT mode) 3	Section 21.4.37
110h	DITCSRA4	Left (even TDM time slot) channel status register (DIT mode) 4	Section 21.4.37
114h	DITCSRA5	Left (even TDM time slot) channel status register (DIT mode) 5	Section 21.4.37
118h	DITCSRB0	Right (odd TDM time slot) channel status register (DIT mode) 0	Section 21.4.38
11Ch	DITCSRB1	Right (odd TDM time slot) channel status register (DIT mode) 1	Section 21.4.38
120h	DITCSRB2	Right (odd TDM time slot) channel status register (DIT mode) 2	Section 21.4.38
124h	DITCSRB3	Right (odd TDM time slot) channel status register (DIT mode) 3	Section 21.4.38
128h	DITCSRB4	Right (odd TDM time slot) channel status register (DIT mode) 4	Section 21.4.38
12Ch	DITCSRB5	Right (odd TDM time slot) channel status register (DIT mode) 5	Section 21.4.38
130h	DITUDRA0	Left (even TDM time slot) channel user data register (DIT mode) 0	Section 21.4.39
134h	DITUDRA1	Left (even TDM time slot) channel user data register (DIT mode) 1	Section 21.4.39
138h	DITUDRA2	Left (even TDM time slot) channel user data register (DIT mode) 2	Section 21.4.39
13Ch	DITUDRA3	Left (even TDM time slot) channel user data register (DIT mode) 3	Section 21.4.39
140h	DITUDRA4	Left (even TDM time slot) channel user data register (DIT mode) 4	Section 21.4.39
144h	DITUDRA5	Left (even TDM time slot) channel user data register (DIT mode) 5	Section 21.4.39
148h	DITUDRB0	Right (odd TDM time slot) channel user data register (DIT mode) 0	Section 21.4.40

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 21-8. McASP Registers Accessed Through Configuration Bus (continued)

Offset	Acronym	Register Description	Section
14Ch	DITUDRB1	Right (odd TDM time slot) channel user data register (DIT mode) 1	Section 21.4.40
150h	DITUDRB2	Right (odd TDM time slot) channel user data register (DIT mode) 2	Section 21.4.40
154h	DITUDRB3	Right (odd TDM time slot) channel user data register (DIT mode) 3	Section 21.4.40
158h	DITUDRB4	Right (odd TDM time slot) channel user data register (DIT mode) 4	Section 21.4.40
15Ch	DITUDRB5	Right (odd TDM time slot) channel user data register (DIT mode) 5	Section 21.4.40
180h	SRCTL0	Serializer control register 0	Section 21.4.36
184h	SRCTL1	Serializer control register 1	Section 21.4.36
188h	SRCTL2	Serializer control register 2	Section 21.4.36
18Ch	SRCTL3	Serializer control register 3	Section 21.4.36
190h	SRCTL4	Serializer control register 4	Section 21.4.36
194h	SRCTL5	Serializer control register 5	Section 21.4.36
200h	XBUF0	Transmit buffer register for serializer 0	Section 21.4.41
204h	XBUF1	Transmit buffer register for serializer 1	Section 21.4.41
208h	XBUF2	Transmit buffer register for serializer 2	Section 21.4.41
20Ch	XBUF3	Transmit buffer register for serializer 3	Section 21.4.41
210h	XBUF4	Transmit buffer register for serializer 4	Section 21.4.41
214h	XBUF5	Transmit buffer register for serializer 5	Section 21.4.41
280h	RBUF0	Receive buffer register for serializer 0	Section 21.4.42
284h	RBUF1	Receive buffer register for serializer 1	Section 21.4.42
288h	RBUF2	Receive buffer register for serializer 2	Section 21.4.42
28Ch	RBUF3	Receive buffer register for serializer 3	Section 21.4.42
290h	RBUF4	Receive buffer register for serializer 4	Section 21.4.42
294h	RBUF5	Receive buffer register for serializer 5	Section 21.4.42

Table 21-9. McASP Registers Accessed Through Data Port

Hex Address	Register Name	Register Description
Read Accesses	RBUF	Receive buffer data port address. Cycles through receive serializers, skipping over transmit serializers and inactive serializers. Starts at the lowest serializer at the beginning of each time slot. DAT BUS only if XBUSEL = 0.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 21-9. McASP Registers Accessed Through Data Port (continued)

Hex Address	Register Name	Register Description
Write Accesses	XBUF	Transmit buffer data port address. Cycles through transmit serializers, skipping over receive and inactive serializers. Starts at the lowest serializer at the beginning of each time slot. DAT BUS only if RBUSEL = 0.

Table 21-10. McASP AFIFO Registers Accessed Through Peripheral Configuration Port

Offset	Acronym	Register Description	Section
1000h	WFIFOCTL	Write FIFO control register	Section 21.4.43
1004h	WFIFOSTS	Write FIFO status register	Section 21.4.44
1008h	RFIFOCTL	Read FIFO control register	Section 21.4.45
100Ch	RFIFOSTS	Read FIFO status register	Section 21.4.46

21.4.1 Revision Identification Register (REV)

The revision identification register (REV) contains identification data for the peripheral. The REV is shown in Figure 21-24 and described in Table 21-11.

Figure 21-24. Revision Identification Register (REV)

31	0
REV	
R-4430 0A02h	

LEGEND: R = Read only; -*n* = value after reset

Table 21-11. Revision Identification Register (REV) Field Descriptions

Bit	Field	Value	Description
31-0	REV	4430 0A02h	Identifies revision of peripheral.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.2 Pin Function Register (PFUNC)

The pin function register (PFUNC) specifies the function of AXRn, ACLKX, AHCLKX, AFSX, ACLKR, AHCLKR, and AFSR pins as either a McASP pin or a general-purpose input/output (GPIO) pin. The PFUNC is shown in Figure 21-25 and described in Table 21-12.

CAUTION

Writing to Reserved Bits

Writing a value other than 0 to reserved bits in this register may cause improper device operation.

Figure 21-25. Pin Function Register (PFUNC)

31	30	29	28	27	26	25	24
AFSR	AHCLKR	ACLKR	AFSX	AHCLKX	ACLKX	AMUTE	Reserved
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0

23	8
Reserved	
R-0	

7	6	5	4	3	2	1	0
Reserved		AXR5	AXR4	AXR3	AXR2	AXR1	AXR0
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 21-12. Pin Function Register (PFUNC) Field Descriptions

Bit	Field	Value	Description
31	AFSR		Determines if AFSR pin functions as McASP or GPIO.
		0	Pin functions as McASP pin.
		1	Pin functions as GPIO pin.
30	AHCLKR		Determines if AHCLKR pin functions as McASP or GPIO.
		0	Pin functions as McASP pin.
		1	Pin functions as GPIO pin.
29	ACLKR		Determines if ACLKR pin functions as McASP or GPIO.
		0	Pin functions as McASP pin.
		1	Pin functions as GPIO pin.
28	AFSX		Determines if AFSX pin functions as McASP or GPIO.
		0	Pin functions as McASP pin.
		1	Pin functions as GPIO pin.
27	AHCLKX		Determines if AHCLKX pin functions as McASP or GPIO.
		0	Pin functions as McASP pin.
		1	Pin functions as GPIO pin.
26	ACLKX		Determines if ACLKX pin functions as McASP or GPIO.
		0	Pin functions as McASP pin.
		1	Pin functions as GPIO pin.
25	AMUTE		Determines if AMUTE pin functions as McASP or GPIO.
		0	Pin functions as McASP pin.
		1	Pin functions as GPIO pin.
24-6	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
5-0	AXR[5-0]		Determines if AXRn pin functions as McASP or GPIO.
		0	Pin functions as McASP pin.
		1	Pin functions as GPIO pin.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.3 Pin Direction Register (PDIR)

The pin direction register (PDIR) specifies the direction of AXRn, ACLKX, AHCLKX, AFSX, ACLKR, AHCLKR, and AFSR pins as either an input or an output pin. The PDIR is shown in Figure 21-26 and described in Table 21-13.

Regardless of the pin function register (PFUNC) setting, each PDIR bit must be set to 1 for the specified pin to be enabled as an output and each PDIR bit must be cleared to 0 for the specified pin to be an input.

For example, if the McASP is configured to use an internally-generated bit clock and the clock is to be driven out to the system, the PFUNC bit must be cleared to 0 (McASP function) and the PDIR bit must be set to 1 (an output).

When AXRn is configured to transmit, the PFUNC bit must be cleared to 0 (McASP function) and the PDIR bit must be set to 1 (an output). Similarly, when AXRn is configured to receive, the PFUNC bit must be cleared to 0 (McASP function) and the PDIR bit must be cleared to 0 (an input).

CAUTION

Writing to Reserved Bits

Writing a value other than 0 to reserved bits in this register may cause improper device operation.

Figure 21-26. Pin Direction Register (PDIR)

31	30	29	28	27	26	25	24
AFSR	AHCLKR	ACLKR	AFSX	AHCLKX	ACLKX	AMUTE	Reserved
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0

23	8
Reserved	
R-0	

7	6	5	4	3	2	1	0
Reserved		AXR5	AXR4	AXR3	AXR2	AXR1	AXR0
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 21-13. Pin Direction Register (PDIR) Field Descriptions

Bit	Field	Value	Description
31	AFSR		Determines if AFSR pin functions as an input or output.
		0	Pin functions as input.
		1	Pin functions as output.
30	AHCLKR		Determines if AHCLKR pin functions as an input or output.
		0	Pin functions as input.
		1	Pin functions as output.
29	ACLKR		Determines if ACLKR pin functions as an input or output.
		0	Pin functions as input.
		1	Pin functions as output.
28	AFSX		Determines if AFSX pin functions as an input or output.
		0	Pin functions as input.
		1	Pin functions as output.
27	AHCLKX		Determines if AHCLKX pin functions as an input or output.
		0	Pin functions as input.
		1	Pin functions as output.
26	ACLKX		Determines if ACLKX pin functions as an input or output.
		0	Pin functions as input.
		1	Pin functions as output.
25	AMUTE		Determines if AMUTE pin functions as an input or output.
		0	Pin functions as input.
		1	Pin functions as output.
24-6	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
5-0	AXR[5-0]		Determines if AXRn pin functions as an input or output.
		0	Pin functions as input.
		1	Pin functions as output.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.4 Pin Data Output Register (PDOUT)

The pin data output register (PDOUT) holds a value for data out at all times, and may be read back at all times. The value held by PDOUT is not affected by writing to PDIR and PFUNC. However, the data value in PDOUT is driven out onto the McASP pin only if the corresponding bit in PFUNC is set to 1 (GPIO function) and the corresponding bit in PDIR is set to 1 (output). When reading data, returns the corresponding bit value in PDOUT[n], does not return input from I/O pin; when writing data, writes to the corresponding PDOUT[n] bit. The PDOUT is shown in Figure 21-27 and described in Table 21-14.

PDOUT has these aliases or alternate addresses:

- PDSET - when written to at this address, writing a 1 to a bit in PDSET sets the corresponding bit in PDOUT to 1; writing a 0 has no effect and keeps the bits in PDOUT unchanged.
- PDCLR - when written to at this address, writing a 1 to a bit in PDCLR clears the corresponding bit in PDOUT to 0; writing a 0 has no effect and keeps the bits in PDOUT unchanged.

There is only one set of data out bits, PDOUT[31-0]. The other registers, PDSET and PDCLR, are just different addresses for the same control bits, with different behaviors during writes.

CAUTION

Writing to Reserved Bits

Writing a value other than 0 to reserved bits in this register may cause improper device operation.

Figure 21-27. Pin Data Output Register (PDOUT)

31	30	29	28	27	26	25	24
AFSR	AHCLKR	ACLKR	AFSX	AHCLKX	ACLKX	AMUTE	Reserved
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0

23	8
Reserved	
R-0	

7	6	5	4	3	2	1	0
Reserved		AXR5	AXR4	AXR3	AXR2	AXR1	AXR0
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 21-14. Pin Data Output Register (PDOUT) Field Descriptions

Bit	Field	Value	Description
31	AFSR		Determines drive on AFSR output pin when the corresponding PFUNC[31] and PDIR[31] bits are set to 1.
		0	Pin drives low.
		1	Pin drives high.
30	AHCLKR		Determines drive on AHCLKR output pin when the corresponding PFUNC[30] and PDIR[30] bits are set to 1.
		0	Pin drives low.
		1	Pin drives high.
29	ACLKR		Determines drive on ACLKR output pin when the corresponding PFUNC[29] and PDIR[29] bits are set to 1.
		0	Pin drives low.
		1	Pin drives high.
28	AFSX		Determines drive on AFSX output pin when the corresponding PFUNC[28] and PDIR[28] bits are set to 1.
		0	Pin drives low.
		1	Pin drives high.
27	AHCLKX		Determines drive on AHCLKX output pin when the corresponding PFUNC[27] and PDIR[27] bits are set to 1.
		0	Pin drives low.
		1	Pin drives high.
26	ACLKX		Determines drive on ACLKX output pin when the corresponding PFUNC[26] and PDIR[26] bits are set to 1.
		0	Pin drives low.
		1	Pin drives high.
25	AMUTE		Determines drive on AMUTE output pin when the corresponding PFUNC[25] and PDIR[25] bits are set to 1.
		0	Pin drives low.
		1	Pin drives high.
24-6	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
5-0	AXR[5-0]		Determines drive on AXR[n] output pin when the corresponding PFUNC[n] and PDIR[n] bits are set to 1.
		0	Pin drives low.
		1	Pin drives high.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.5 Pin Data Input Register (PDIN)

The pin data input register (PDIN) holds the I/O pin state of each of the McASP pins. PDIN allows the actual value of the pin to be read, regardless of the state of PFUNC and PDIR. The value after reset for registers 1 through 15 and 24 through 31 depends on how the pins are being driven. The PDIN is shown in Figure 21-28 and described in Table 21-15.

CAUTION

Writing to Reserved Bits

Writing a value other than 0 to reserved bits in this register may cause improper device operation.

Figure 21-28. Pin Data Input Register (PDIN)

31	30	29	28	27	26	25	24
AFSR	AHCLKR	ACLKR	AFSX	AHCLKX	ACLKX	AMUTE	Reserved
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0

23	8
Reserved	
R-0	

7	6	5	4	3	2	1	0
Reserved		AXR5	AXR4	AXR3	AXR2	AXR1	AXR0
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 21-15. Pin Data Input Register (PDIN) Field Descriptions

Bit	Field	Value	Description
31	AFSR		Logic level on AFSR pin.
		0	Pin is logic low.
		1	Pin is logic high.
30	AHCLKR		Logic level on AHCLKR pin.
		0	Pin is logic low.
		1	Pin is logic high.
29	ACLKR		Logic level on ACLKR pin.
		0	Pin is logic low.
		1	Pin is logic high.
28	AFSX		Logic level on AFSX pin.
		0	Pin is logic low.
		1	Pin is logic high.
27	AHCLKX		Logic level on AHCLKX pin.
		0	Pin is logic low.
		1	Pin is logic high.
26	ACLKX		Logic level on ACLKX pin.
		0	Pin is logic low.
		1	Pin is logic high.
25	AMUTE		Logic level on AMUTE pin.
		0	Pin is logic low.
		1	Pin is logic high.
24-6	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
5-0	AXR[5-0]		Logic level on AXR[n] pin.
		0	Pin is logic low.
		1	Pin is logic high.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.6 Pin Data Set Register (PDSET)

The pin data set register (PDSET) is an alias of the pin data output register (PDOUT) for writes only. Writing a 1 to the PDSET bit sets the corresponding bit in PDOUT and, if PFUNC = 1 (GPIO function) and PDIR = 1 (output), drives a logic high on the pin. PDSET is useful for a multitasking system because it allows you to set to a logic high only the desired pin(s) within a system without affecting other I/O pins controlled by the same McASP. The PDSET is shown in Figure 21-29 and described in Table 21-16.

CAUTION

Writing to Reserved Bits

Writing a value other than 0 to reserved bits in this register may cause improper device operation.

Figure 21-29. Pin Data Set Register (PDSET)

31	30	29	28	27	26	25	24
AFSR	AHCLKR	ACLKR	AFSX	AHCLKX	ACLKX	AMUTE	Reserved
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0

23	8
Reserved	
R-0	

7	6	5	4	3	2	1	0
Reserved		AXR5	AXR4	AXR3	AXR2	AXR1	AXR0
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 21-16. Pin Data Set Register (PDSET) Field Descriptions

Bit	Field	Value	Description
31	AFSR		Allows the corresponding AFSR bit in PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[31] bit is set to 1.
30	AHCLKR		Allows the corresponding AHCLKR bit in PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[30] bit is set to 1.
29	ACLKR		Allows the corresponding ACLKR bit in PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[29] bit is set to 1.
28	AFSX		Allows the corresponding AFSX bit in PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[28] bit is set to 1.
27	AHCLKX		Allows the corresponding AHCLKX bit in PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[27] bit is set to 1.
26	ACLKX		Allows the corresponding ACLKX bit in PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[26] bit is set to 1.
25	AMUTE		Allows the corresponding AMUTE bit in PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[25] bit is set to 1.
24-6	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
5-0	AXR[5-0]		Allows the corresponding AXR[n] bit in PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[n] bit is set to 1.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.7 Pin Data Clear Register (PDCLR)

The pin data clear register (PDCLR) is an alias of the pin data output register (PDOUT) for writes only. Writing a 1 to the PDCLR bit clears the corresponding bit in PDOUT and, if PFUNC = 1 (GPIO function) and PDIR = 1 (output), drives a logic low on the pin. PDCLR is useful for a multitasking system because it allows you to clear to a logic low only the desired pin(s) within a system without affecting other I/O pins controlled by the same McASP. The PDCLR is shown in Figure 21-30 and described in Table 21-17.

CAUTION

Writing to Reserved Bits

Writing a value other than 0 to reserved bits in this register may cause improper device operation.

Figure 21-30. Pin Data Clear Register (PDCLR)

31	30	29	28	27	26	25	24
AFSR	AHCLKR	ACLKR	AFSX	AHCLKX	ACLKX	AMUTE	Reserved
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0

23	8
Reserved	
R-0	

7	6	5	4	3	2	1	0
Reserved		AXR5	AXR4	AXR3	AXR2	AXR1	AXR0
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 21-17. Pin Data Clear Register (PDCLR) Field Descriptions

Bit	Field	Value	Description
31	AFSR		Allows the corresponding AFSR bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[31] bit is cleared to 0.
30	AHCLKR		Allows the corresponding AHCLKR bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[30] bit is cleared to 0.
29	ACLKR		Allows the corresponding ACLKR bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[29] bit is cleared to 0.
28	AFSX		Allows the corresponding AFSX bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[28] bit is cleared to 0.
27	AHCLKX		Allows the corresponding AHCLKX bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[27] bit is cleared to 0.
26	ACLKX		Allows the corresponding ACLKX bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[26] bit is cleared to 0.
25	AMUTE		Allows the corresponding AMUTE bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[25] bit is cleared to 0.
24-6	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
5-0	AXR[5-0]		Allows the corresponding AXR[n] bit in PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port.
		0	No effect.
		1	PDOUT[n] bit is cleared to 0.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.8 Global Control Register (GBLCTL)

The global control register (GBLCTL) provides initialization of the transmit and receive sections. The GBLCTL is shown in Figure 21-31 and described in Table 21-18.

The bit fields in GBLCTL are synchronized and latched by the corresponding clocks (ACLKX for bits 12-8 and ACLKR for bits 4-0). Before GBLCTL is programmed, you must ensure that serial clocks are running. If the corresponding external serial clocks, ACLKX and ACLKR, are not yet running, you should select the internal serial clock source in AHCLKXCTL, AHCLKRCTL, ACLKXCTL, and ACLKRCTL before GBLCTL is programmed. Also, after programming any bits in GBLCTL you should not proceed until you have read back from GBLCTL and verified that the bits are latched in GBLCTL.

Figure 21-31. Global Control Register (GBLCTL)

31						16
Reserved						
R-0						
15	13	12	11	10	9	8
Reserved		XFRST	XSMRST	XSRCLR	XHCLKRST	XCLKRST
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7	5	4	3	2	1	0
Reserved		RFRST	RSMRST	RSRCLR	RHCLKRST	RCLKRST
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 21-18. Global Control Register (GBLCTL) Field Descriptions

Bit	Field	Value	Description
31-13	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
12	XFRST		Transmit frame sync generator reset enable bit.
		0	Transmit frame sync generator is reset.
		1	Transmit frame sync generator is active. When released from reset, the transmit frame sync generator begins counting serial clocks and generating frame sync as programmed.
11	XSMRST		Transmit state machine reset enable bit.
		0	Transmit state machine is held in reset. AXRn pin state: If PFUNC[n] = 0 and PDIR[n] = 1; then the serializer drives the AXRn pin to the state specified for inactive time slot (as determined by DISMOD bits in SRCTL).
		1	Transmit state machine is released from reset. When released from reset, the transmit state machine immediately transfers data from XRBUF[n] to XRSR[n]. The transmit state machine sets the underrun flag (XUNDRN) in XSTAT, if XRBUF[n] have not been preloaded with data before reset is released. The transmit state machine also immediately begins detecting frame sync and is ready to transmit. Transmit TDM time slot begins at slot 0 after reset is released.
10	XSRCLR		Transmit serializer clear enable bit. By clearing then setting this bit, the transmit buffer is flushed to an empty state (XDATA = 1). If XSMRST = 1, XSRCLR = 1, XDATA = 1, and XBUF is not loaded with new data before the start of the next active time slot, an underrun will occur.
		0	Transmit serializers are cleared.
		1	Transmit serializers are active. When the transmit serializers are first taken out of reset (XSRCLR changes from 0 to 1), the transmit data ready bit (XDATA) in XSTAT is set to indicate XBUF is ready to be written.
9	XHCLKRST		Transmit high-frequency clock divider reset enable bit.
		0	Transmit high-frequency clock divider is held in reset and passes through its input as divide-by-1..
		1	Transmit high-frequency clock divider is running.
8	XCLKRST		Transmit clock divider reset enable bit.
		0	Transmit clock divider is held in reset. When the clock divider is in reset, it passes through a divide-by-1 of its input.
		1	Transmit clock divider is running.

Copyright © 2011, Texas Instruments Incorporated

Table 21-18. Global Control Register (GBLCTL) Field Descriptions (continued)

Bit	Field	Value	Description
7-5	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
4	RFRST		Receive frame sync generator reset enable bit.
		0	Receive frame sync generator is reset.
		1	Receive frame sync generator is active. When released from reset, the receive frame sync generator begins counting serial clocks and generating frame sync as programmed.
3	RSMRST		Receive state machine reset enable bit.
		0	Receive state machine is held in reset.
		1	Receive state machine is released from reset. When released from reset, the receive state machine immediately begins detecting frame sync and is ready to receive. Receive TDM time slot begins at slot 0 after reset is released.
2	RSRCLR		Receive serializer clear enable bit. By clearing then setting this bit, the receive buffer is flushed.
		0	Receive serializers are cleared.
		1	Receive serializers are active.
1	RHCLKRST		Receive high-frequency clock divider reset enable bit.
		0	Receive high-frequency clock divider is held in reset and passes through its input as divide-by-1.
		1	Receive high-frequency clock divider is running.
0	RCLKRST		Receive high-frequency clock divider reset enable bit.
		0	Receive clock divider is held in reset. When the clock divider is in reset, it passes through a divide-by-1 of its input.
		1	Receive clock divider is running.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.9 Audio Mute Control Register (AMUTE)

The audio mute control register (AMUTE) controls the McASP audio mute (AMUTE) output pin. The value after reset for register 4 depends on how the pins are being driven. The AMUTE is shown in Figure 21-32 and described in Table 21-19.

Figure 21-32. Audio Mute Control Register (AMUTE)

31						16
Reserved						
R-0						

15		13	12	11	10	9	8
Reserved			XDMAERR	RDMAERR	XCKFAIL	RCKFAIL	XSYNCERR
R-0			R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

7	6	5	4	3	2	1	0
RSYNCERR	XUNDRN	ROVRN	INSTAT	INEN	INPOL	MUTEN	
R/W-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 21-19. Audio Mute Control Register (AMUTE) Field Descriptions

Bit	Field	Value	Description
31-13	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
12	XDMAERR		If transmit DMA error (XDMAERR), drive AMUTE active enable bit.
		0	Drive is disabled. Detection of transmit DMA error is ignored by AMUTE.
		1	Drive is enabled (active). Upon detection of transmit DMA error, AMUTE is active and is driven according to MUTEN bit.
11	RDMAERR		If receive DMA error (RDMAERR), drive AMUTE active enable bit.
		0	Drive is disabled. Detection of receive DMA error is ignored by AMUTE.
		1	Drive is enabled (active). Upon detection of receive DMA error, AMUTE is active and is driven according to MUTEN bit.
10	XCKFAIL		If transmit clock failure (XCKFAIL), drive AMUTE active enable bit.
		0	Drive is disabled. Detection of transmit clock failure is ignored by AMUTE.
		1	Drive is enabled (active). Upon detection of transmit clock failure, AMUTE is active and is driven according to MUTEN bit
9	RCKFAIL		If receive clock failure (RCKFAIL), drive AMUTE active enable bit.
		0	Drive is disabled. Detection of receive clock failure is ignored by AMUTE.
		1	Drive is enabled (active). Upon detection of receive clock failure, AMUTE is active and is driven according to MUTEN bit.
8	XSYNCERR		If unexpected transmit frame sync error (XSYNCERR), drive AMUTE active enable bit.
		0	Drive is disabled. Detection of unexpected transmit frame sync error is ignored by AMUTE.
		1	Drive is enabled (active). Upon detection of unexpected transmit frame sync error, AMUTE is active and is driven according to MUTEN bit.
7	RSYNCERR		If unexpected receive frame sync error (RSYNCERR), drive AMUTE active enable bit.
		0	Drive is disabled. Detection of unexpected receive frame sync error is ignored by AMUTE.
		1	Drive is enabled (active). Upon detection of unexpected receive frame sync error, AMUTE is active and is driven according to MUTEN bit.
6	XUNDRN		If transmit underrun error (XUNDRN), drive AMUTE active enable bit.
		0	Drive is disabled. Detection of transmit underrun error is ignored by AMUTE.
		1	Drive is enabled (active). Upon detection of transmit underrun error, AMUTE is active and is driven according to MUTEN bit.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 21-19. Audio Mute Control Register (AMUTE) Field Descriptions (continued)

Bit	Field	Value	Description
5	ROVRN		If receiver overrun error (ROVRN), drive AMUTE active enable bit.
		0	Drive is disabled. Detection of receiver overrun error is ignored by AMUTE.
		1	Drive is enabled (active). Upon detection of receiver overrun error, AMUTE is active and is driven according to MUTEN bit.
4	INSTAT		Determines drive on AXRn pin when PFUNC[n] and PDIR[n] bits are set to 1.
		0	AMUTEIN pin is inactive.
		1	AMUTEIN pin is active. Audio mute in error is detected.
3	INEN		Drive AMUTE active when AMUTEIN error is active (INSTAT = 1).
		0	Drive is disabled. AMUTEIN is ignored by AMUTE.
		1	Drive is enabled (active). INSTAT = 1 drives AMUTE active.
2	INPOL		Audio mute in (AMUTEIN) polarity select bit.
		0	Polarity is active high. A high on AMUTEIN sets INSTAT to 1.
		1	Polarity is active low. A low on AM UTEIN sets INSTAT to 1.
1-0	MUTEN	0-3h	AMUTE pin enable bit (unless overridden by GPIO registers).
		0	AMUTE pin is disabled, pin goes to tri-state condition.
		1h	AMUTE pin is driven high if error is detected.
		2h	AMUTE pin is driven low if error is detected.
		3h	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.10 Digital Loopback Control Register (DLBCTL)

The digital loopback control register (DLBCTL) controls the internal loopback settings of the McASP in TDM mode. The DLBCTL is shown in Figure 21-33 and described in Table 21-20.

Figure 21-33. Digital Loopback Control Register (DLBCTL)

31															16
Reserved															
R-0															

15												4	3	2	1	0
Reserved													MODE		ORD	DLBEN
R-0													R/W-0		R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 21-20. Digital Loopback Control Register (DLBCTL) Field Descriptions

Bit	Field	Value	Description
31-4	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
3-2	MODE	0-3h	Loopback generator mode bits. Applies only when loopback mode is enabled (DLBEN = 1).
		0	Default and reserved on loopback mode (DLBEN = 1). When in non-loopback mode (DLBEN = 0), MODE should be left at default (00). When in loopback mode (DLBEN = 1), MODE = 00 is reserved and is not applicable.
		1h	Transmit clock and frame sync generators used by both transmit and receive sections. When in loopback mode (DLBEN = 1), MODE must be 01.
		2h-3h	Reserved.
1	ORD		Loopback order bit when loopback mode is enabled (DLBEN = 1).
		0	Odd serializers N + 1 transmit to even serializers N that receive. The corresponding serializers must be programmed properly.
		1	Even serializers N transmit to odd serializers N + 1 that receive. The corresponding serializers must be programmed properly.
0	DLBEN		Loopback mode enable bit.
		0	Loopback mode is disabled.
		1	Loopback mode is enabled.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.11 Digital Mode Control Register (DITCTL)

The DIT mode control register (DITCTL) controls DIT operations of the McASP. The DITCTL is shown in Figure 21-34 and described in Table 21-21.

Figure 21-34. Digital Mode Control Register (DITCTL)

31					16
Reserved					
R-0					

15	4	3	2	1	0
Reserved		VB	VA	Rsvd	DITEN
R-0		R/W-0	R/W-0	R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 21-21. Digital Mode Control Register (DITCTL) Field Descriptions

Bit	Field	Value	Description
31-4	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
3	VB		Valid bit for odd time slots (DIT right subframe).
		0	V bit is 0 during odd DIT subframes.
		1	V bit is 1 during odd DIT subframes.
2	VA		Valid bit for even time slots (DIT left subframe).
		0	V bit is 0 during even DIT subframes.
		1	V bit is 1 during even DIT subframes.
1	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
0	DITEN		DIT mode enable bit. DITEN should only be changed while the XSMRST bit in GBLCTL is in reset (and for startup, XSRCLR also in reset). However, it is not necessary to reset the XCLKRST or XHCLKRST bits in GBLCTL to change DITEN.
		0	DIT mode is disabled. Transmitter operates in TDM or burst mode.
		1	DIT mode is enabled. Transmitter operates in DIT encoded mode.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.12 Receiver Global Control Register (RGBLCTL)

Alias of the global control register (GBLCTL). Writing to the receiver global control register (RGBLCTL) affects only the receive bits of GBLCTL (bits 4-0). Reads from RGBLCTL return the value of GBLCTL. RGBLCTL allows the receiver to be reset independently from the transmitter. The RGBLCTL is shown in Figure 21-35 and described in Table 21-22. See Section 21.4.8 for a detailed description of GBLCTL.

Figure 21-35. Receiver Global Control Register (RGBLCTL)

31						16
Reserved						
R-0						
15	13	12	11	10	9	8
Reserved		XFRST	XSMRST	XSRCLR	XHCLKRST	XCLKRST
R-0		R-0	R-0	R-0	R-0	R-0
7	5	4	3	2	1	0
Reserved		RFRST	RSMRST	RSRCLR	RHCLKRST	RCLKRST
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 21-22. Receiver Global Control Register (RGBLCTL) Field Descriptions

Bit	Field	Value	Description
31-13	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
12	XFRST	x	Transmit frame sync generator reset enable bit. A read of this bit returns the XFRST bit value of GBLCTL. Writes have no effect.
11	XSMRST	x	Transmit state machine reset enable bit. A read of this bit returns the XSMRST bit value of GBLCTL. Writes have no effect.
10	XSRCLR	x	Transmit serializer clear enable bit. A read of this bit returns the XSRCLR bit value of GBLCTL. Writes have no effect.
9	XHCLKRST	x	Transmit high-frequency clock divider reset enable bit. A read of this bit returns the XHCLKRST bit value of GBLCTL. Writes have no effect.
8	XCLKRST	x	Transmit clock divider reset enable bit. A read of this bit returns the XCLKRST bit value of GBLCTL. Writes have no effect.
7-5	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
4	RFRST		Receive frame sync generator reset enable bit. A write to this bit affects the RFRST bit of GBLCTL.
		0	Receive frame sync generator is reset.
		1	Receive frame sync generator is active.
3	RSMRST		Receive state machine reset enable bit. A write to this bit affects the RSMRST bit of GBLCTL.
		0	Receive state machine is held in reset.
		1	Receive state machine is released from reset.
2	RSRCLR		Receive serializer clear enable bit. A write to this bit affects the RSRCLR bit of GBLCTL.
		0	Receive serializers are cleared.
		1	Receive serializers are active.
1	RHCLKRST		Receive high-frequency clock divider reset enable bit. A write to this bit affects the RHCLKRST bit of GBLCTL.
		0	Receive high-frequency clock divider is held in reset and passes through its input as divide-by-1.
		1	Receive high-frequency clock divider is running.
0	RCLKRST		Receive clock divider reset enable bit. A write to this bit affects the RCLKRST bit of GBLCTL.
		0	Receive clock divider is held in reset.
		1	Receive clock divider is running.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.13 Receive Format Unit Bit Mask Register (RMASK)

The receive format unit bit mask register (RMASK) determines which bits of the received data are masked off and padded with a known value before being read by the CPU or DMA. The RMASK is shown in Figure 21-36 and described in Table 21-23.

Figure 21-36. Receive Format Unit Bit Mask Register (RMASK)

31	0
RMASK[*n*]	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 21-23. Receive Format Unit Bit Mask Register (RMASK) Field Descriptions

Bit	Field	Value	Description
31-0	RMASK[31-0]		Receive data mask *n* enable bit.
		0	Corresponding bit of receive data (after passing through reverse and rotate units) is masked out and then padded with the selected bit pad value (RPAD and RPBIT bits in RFMT).
		1	Corresponding bit of receive data (after passing through reverse and rotate units) is returned to CPU or DMA.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.14 Receive Bit Stream Format Register (RFMT)

The receive bit stream format register (RFMT) configures the receive data format. The RFMT is shown in Figure 21-37 and described in Table 21-24.

Figure 21-37. Receive Bit Stream Format Register (RFMT)

31–18	17–16
Reserved	RDATDLY
R-0	R/W-0

15	14–13	12–8	7–4	3	2–0
RRVRS	RPAD	RPBIT	RSSZ	RBUSEL	RROT
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 21-24. Receive Bit Stream Format Register (RFMT) Field Descriptions

Bit	Field	Value	Description
31-18	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
17-16	RDATDLY	0-3h	Receive bit delay.
		0	0-bit delay. The first receive data bit, AXRn, occurs in same ACLKR cycle as the receive frame sync (AFSR).
		1h	1-bit delay. The first receive data bit, AXRn, occurs one ACLKR cycle after the receive frame sync (AFSR).
		2h	2-bit delay. The first receive data bit, AXRn, occurs two ACLKR cycles after the receive frame sync (AFSR).
		3h	Reserved.
15	RRVRS		Receive serial bitstream order.
		0	Bitstream is LSB first. No bit reversal is performed in receive format bit reverse unit.
		1	Bitstream is MSB first. Bit reversal is performed in receive format bit reverse unit.
14-13	RPAD	0-3h	Pad value for extra bits in slot not belonging to the word. This field only applies to bits when RMASK[n] = 0.
		0	Pad extra bits with 0.
		1h	Pad extra bits with 1.
		2h	Pad extra bits with one of the bits from the word as specified by RPBIT bits.
		3h	Reserved.
12-8	RPBIT	0-1Fh	RPBIT value determines which bit (as read by the CPU or DMA from RBUF[n]) is used to pad the extra bits. This field only applies when RPAD = 2h.
		0	Pad with bit 0 value.
		1h-1Fh	Pad with bit 1 to bit 31 value.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 21-24. Receive Bit Stream Format Register (RFMT) Field Descriptions (continued)

Bit	Field	Value	Description
7-4	RSSZ	0-Fh	Receive slot size.
		0-2h	Reserved
		3h	Slot size is 8 bits.
		4h	Reserved
		5h	Slot size is 12 bits.
		6h	Reserved
		7h	Slot size is 16 bits.
		8h	Reserved
		9h	Slot size is 20 bits.
		Ah	Reserved
		Bh	Slot size is 24 bits
		Ch	Reserved
		Dh	Slot size is 28 bits.
		Eh	Reserved
		Fh	Slot size is 32 bits.
3	RBUSEL		Selects whether reads from serializer buffer XRBUF[n] originate from the configuration bus (CFG) or the data (DAT) port.
		0	Reads from XRBUF[n] originate on data port. Reads from XRBUF[n] on configuration bus are ignored.
		1	Reads from XRBUF[n] originate on configuration bus. Reads from XRBUF[n] on data port are ignored.
2-0	RROT	0-7h	Right-rotation value for receive rotate right format unit.
		0	Rotate right by 0 (no rotation).
		1h	Rotate right by 4 bit positions.
		2h	Rotate right by 8 bit positions.
		3h	Rotate right by 12 bit positions.
		4h	Rotate right by 16 bit positions.
		5h	Rotate right by 20 bit positions.
		6h	Rotate right by 24 bit positions.
		7h	Rotate right by 28 bit positions.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.15 Receive Frame Sync Control Register (AFSRCTL)

The receive frame sync control register (AFSRCTL) configures the receive frame sync (AFSR). The AFSRCTL is shown in Figure 21-38 and described in Table 21-25.

Figure 21-38. Receive Frame Sync Control Register (AFSRCTL)

31							16
Reserved							
R-0							

15	7	6	5	4	3	2	1	0
RMOD		Reserved		FRWID	Reserved		FSRM	FSRP
R/W-0		R-0		R/W-0	R-0		R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 21-25. Receive Frame Sync Control Register (AFSRCTL) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
15-7	RMOD	0-1FFh	Receive frame sync mode select bits.
		0	Burst mode.
		1h	Reserved.
		2h-20h	2-slot TDM (I2S mode) to 32-slot TDM.
		21h-17Fh	Reserved.
		180h	384-slot TDM (external DIR IC inputting 384-slot DIR frames to McASP over I2S interface).
		181h-1FFh	Reserved.
6-5	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
4	FRWID		Receive frame sync width select bit indicates the width of the receive frame sync (AFSR) during its active period.
		0	Single bit.
		1	Single word.
3-2	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
1	FSRM		Receive frame sync generation select bit.
		0	Externally-generated receive frame sync.
		1	Internally-generated receive frame sync.
0	FSRP		Receive frame sync polarity select bit.
		0	A rising edge on receive frame sync (AFSR) indicates the beginning of a frame.
		1	A falling edge on receive frame sync (AFSR) indicates the beginning of a frame.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.16 Receive Clock Control Register (ACLKRCTL)

The receive clock control register (ACLKRCTL) configures the receive bit clock (ACLKR) and the receive clock generator. The ACLKRCTL is shown in Figure 21-39 and described in Table 21-26.

Figure 21-39. Receive Clock Control Register (ACLKRCTL)

31						16
Reserved						
R-0						

15	8	7	6	5	4	0
Reserved		CLKRP	Rsvd	CLKRM	CLKRDIV	
R-0		R/W-0	R-0	R/W-1	R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 21-26. Receive Clock Control Register (ACLKRCTL) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
7	CLKRP		Receive bitstream clock polarity select bit.
		0	Falling edge. Receiver samples data on the falling edge of the serial clock, so the external transmitter driving this receiver must shift data out on the rising edge of the serial clock.
		1	Rising edge. Receiver samples data on the rising edge of the serial clock, so the external transmitter driving this receiver must shift data out on the falling edge of the serial clock.
6	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
5	CLKRM		Receive bit clock source bit. Note that this bit does not have any effect, if ACLKXCTL.ASYNC = 0.
		0	External receive clock source from ACLKR pin.
		1	Internal receive clock source from output of programmable bit clock divider.
4-0	CLKRDIV	0-1Fh	Receive bit clock divide ratio bits determine the divide-down ratio from AHCLKR to ACLKR. Note that this bit does not have any effect, if ACLKXCTL.ASYNC = 0.
		0	Divide-by-1.
		1h	Divide-by-2.
		2h-1Fh	Divide-by-3 to divide-by-32.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.17 Receive High-Frequency Clock Control Register (AHCLKRCTL)

The receive high-frequency clock control register (AHCLKRCTL) configures the receive high-frequency master clock (AHCLKR) and the receive clock generator. The AHCLKRCTL is shown in Figure 21-40 and described in Table 21-27.

Figure 21-40. Receive High-Frequency Clock Control Register (AHCLKRCTL)

31–16				15–0
Reserved				
R-0				
15	14	13–12	11–0	
HCLKRM	HCLKRP	Reserved	HCLKRDIV	
R/W-1	R/W-0	R/W-0	R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 21-27. Receive High-Frequency Clock Control Register (AHCLKRCTL) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
15	HCLKRM		Receive high-frequency clock source bit.
		0	External receive high-frequency clock source from AHCLKR pin.
		1	Internal receive high-frequency clock source from output of programmable high clock divider.
14	HCLKRP		Receive bitstream high-frequency clock polarity select bit.
		0	AHCLKR is not inverted before programmable bit clock divider. In the special case where the receive bit clock (ACLKR) is internally generated and the programmable bit clock divider is set to divide-by-1 (CLKRDIV = 0 in ACLKRCTL), AHCLKR is directly passed through to the ACLKR pin.
		1	AHCLKR is inverted before programmable bit clock divider. In the special case where the receive bit clock (ACLKR) is internally generated and the programmable bit clock divider is set to divide-by-1 (CLKRDIV = 0 in ACLKRCTL), AHCLKR is directly passed through to the ACLKR pin.
13-12	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
11-0	HCLKRDIV	0-FFFh	Receive high-frequency clock divide ratio bits determine the divide-down ratio from AUXCLK to AHCLKR.
		0	Divide-by-1.
		1h	Divide-by-2.
		2h-FFFh	Divide-by-3 to divide-by-4096.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.18 Receive TDM Time Slot Register (RTDM)

The receive TDM time slot register (RTDM) specifies which TDM time slot the receiver is active. The RTDM is shown in Figure 21-41 and described in Table 21-28.

Figure 21-41. Receive TDM Time Slot Register (RTDM)

31	0
RTDMS[*n*]	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 21-28. Receive TDM Time Slot Register (RTDM) Field Descriptions

Bit	Field	Value	Description
31-0	RTDMS[31-0]		Receiver mode during TDM time slot *n*.
		0	Receive TDM time slot *n* is inactive. The receive serializer does not shift in data during this slot.
		1	Receive TDM time slot *n* is active. The receive serializer shifts in data during this slot.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.19 Receiver Interrupt Control Register (RINTCTL)

The receiver interrupt control register (RINTCTL) controls generation of the McASP receive interrupt (RINT). When the register bit(s) is set to 1, the occurrence of the enabled McASP condition(s) generates RINT. The RINTCTL is shown in Figure 21-42 and described in Table 21-29. See Section 21.4.20 for a description of the interrupt conditions.

Figure 21-42. Receiver Interrupt Control Register (RINTCTL)

31							8
Reserved							
R-0							
7	6	5	4	3	2	1	0
RSTAFRM	Reserved	RDATA	RLAST	RDMAERR	RCKFAIL	RSYNCERR	ROVRN
R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 21-29. Receiver Interrupt Control Register (RINTCTL) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
7	RSTAFRM		Receive start of frame interrupt enable bit.
		0	Interrupt is disabled. A receive start of frame interrupt does not generate a McASP receive interrupt (RINT).
		1	Interrupt is enabled. A receive start of frame interrupt generates a McASP receive interrupt (RINT).
6	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
5	RDATA		Receive data ready interrupt enable bit.
		0	Interrupt is disabled. A receive data ready interrupt does not generate a McASP receive interrupt (RINT).
		1	Interrupt is enabled. A receive data ready interrupt generates a McASP receive interrupt (RINT).
4	RLAST		Receive last slot interrupt enable bit.
		0	Interrupt is disabled. A receive last slot interrupt does not generate a McASP receive interrupt (RINT).
		1	Interrupt is enabled. A receive last slot interrupt generates a McASP receive interrupt (RINT).
3	RDMAERR		Receive DMA error interrupt enable bit.
		0	Interrupt is disabled. A receive DMA error interrupt does not generate a McASP receive interrupt (RINT).
		1	Interrupt is enabled. A receive DMA error interrupt generates a McASP receive interrupt (RINT).
2	RCKFAIL		Receive clock failure interrupt enable bit.
		0	Interrupt is disabled. A receive clock failure interrupt does not generate a McASP receive interrupt (RINT).
		1	Interrupt is enabled. A receive clock failure interrupt generates a McASP receive interrupt (RINT).
1	RSYNCERR		Unexpected receive frame sync interrupt enable bit.
		0	Interrupt is disabled. An unexpected receive frame sync interrupt does not generate a McASP receive interrupt (RINT).
		1	Interrupt is enabled. An unexpected receive frame sync interrupt generates a McASP receive interrupt (RINT).
0	ROVRN		Receiver overrun interrupt enable bit.
		0	Interrupt is disabled. A receiver overrun interrupt does not generate a McASP receive interrupt (RINT).
		1	Interrupt is enabled. A receiver overrun interrupt generates a McASP receive interrupt (RINT).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.20 Receiver Status Register (RSTAT)

The receiver status register (RSTAT) provides the receiver status and receive TDM time slot number. If the McASP logic attempts to set an interrupt flag in the same cycle that the CPU writes to the flag to clear it, the McASP logic has priority and the flag remains set. This also causes a new interrupt request to be generated. The RSTAT is shown in Figure 21-43 and described in Table 21-30.

Figure 21-43. Receiver Status Register (RSTAT)

31						9	8
Reserved							RERR
R-0							R/W-0
7	6	5	4	3	2	1	0
RDMAERR	RSTAFRM	RDATA	RLAST	RTDMSLOT	RCKFAIL	RSYNCERR	ROVRN
R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R-0	R/W1C-0	R/W1C-0	R/W1C-0

LEGEND: R/W = Read/Write; R = Read only; W1C = Writing a 1 clears this bit; writing a 0 has no effect.; -*n* = value after reset

Table 21-30. Receiver Status Register (RSTAT) Field Descriptions

Bit	Field	Value	Description
31-9	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
8	RERR		RERR bit always returns a logic-OR of:ROVRN | RSYNCERR | RCKFAIL | RDMAERR Allows a single bit to be checked to determine if a receiver error interrupt has occurred.
		0	No errors have occurred.
		1	An error has occurred.
7	RDMAERR		Receive DMA error flag. RDMAERR is set when the CPU or DMA reads more serializers through the data port in a given time slot than were programmed as receivers. Causes a receive interrupt (RINT), if this bit is set and RDMAERR in RINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect.
		0	Receive DMA error did not occur.
		1	Receive DMA error did occur.
6	RSTAFRM		Receive start of frame flag. Causes a receive interrupt (RINT), if this bit is set and RSTAFRM in RINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect.
		0	No new receive frame sync (AFSR) is detected.
		1	A new receive frame sync (AFSR) is detected.
5	RDATA		Receive data ready flag. Causes a receive interrupt (RINT), if this bit is set and RDATA in RINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect.
		0	No new data in RBUF.
		1	Data is transferred from XRSR to RBUF and ready to be serviced by the CPU or DMA. When RDATA is set, it always causes a DMA event (AREVT).
4	RLAST		Receive last slot flag. RLAST is set along with RDATA, if the current slot is the last slot in a frame. Causes a receive interrupt (RINT), if this bit is set and RLAST in RINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect.
		0	Current slot is not the last slot in a frame.
		1	Current slot is the last slot in a frame. RDATA is also set.
3	RTDMSLOT		Returns the LSB of RSLOT. Allows a single read of RSTAT to determine whether the current TDM time slot is even or odd.
		0	Current TDM time slot is odd.
		1	Current TDM time slot is even.
2	RCKFAIL		Receive clock failure flag. RCKFAIL is set when the receive clock failure detection circuit reports an error (see *Clock Failure Detection*). Causes a receive interrupt (RINT), if this bit is set and RCKFAIL in RINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect.
		0	Receive clock failure did not occur.
		1	Receive clock failure did occur.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 21-30. Receiver Status Register (RSTAT) Field Descriptions (continued)

Bit	Field	Value	Description
1	RSYNCERR		Unexpected receive frame sync flag. RSYNCERR is set when a new receive frame sync (AFSR) occurs before it is expected. Causes a receive interrupt (RINT), if this bit is set and RSYNCERR in RINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect.
		0	Unexpected receive frame sync did not occur.
		1	Unexpected receive frame sync did occur.
0	ROVRN		Receiver overrun flag. ROVRN is set when the receive serializer is instructed to transfer data from XRSR to RBUF, but the former data in RBUF has not yet been read by the CPU or DMA. Causes a receive interrupt (RINT), if this bit is set and ROVRN in RINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect.
		0	Receiver overrun did not occur.
		1	Receiver overrun did occur.

21.4.21 Current Receive TDM Time Slot Registers (RSLOT)

The current receive TDM time slot register (RSLOT) indicates the current time slot for the receive data frame. The RSLOT is shown in Figure 21-44 and described in Table 21-31.

Figure 21-44. Current Receive TDM Time Slot Registers (RSLOT)

31–16	15–9	8–0
Reserved	Reserved	RSLOTCNT
R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 21-31. Current Receive TDM Time Slot Registers (RSLOT) Field Descriptions

Bit	Field	Value	Description
31-9	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
8-0	RSLOTCNT	0-17Fh	Current receive time slot count. Legal values: 0 to 383 (17Fh). TDM function is not supported for > 32 time slots. However, TDM time slot counter may count to 383 when used to receive a DIR block (transferred over TDM format).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.22 Receive Clock Check Control Register (RCLKCHK)

The receive clock check control register (RCLKCHK) configures the receive clock failure detection circuit. The RCLKCHK is shown in Figure 21-45 and described in Table 21-32.

Figure 21-45. Receive Clock Check Control Register (RCLKCHK)

31–24	23–16
RCNT	RMAX
R-0	R/W-0

15–8	7–4	3–0
RMIN	Reserved	RPS
R/W-0	R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 21-32. Receive Clock Check Control Register (RCLKCHK) Field Descriptions

Bit	Field	Value	Description
31-24	RCNT	0-FFh	Receive clock count value (from previous measurement). The clock circuit continually counts the number of system clocks for every 32 receive high-frequency master clock (AHCLKR) signals, and stores the count in RCNT until the next measurement is taken.
23-16	RMAX	0-FFh	Receive clock maximum boundary. This 8-bit unsigned value sets the maximum allowed boundary for the clock check counter after 32 receive high-frequency master clock (AHCLKR) signals have been received. If the current counter value is greater than RMAX after counting 32 AHCLKR signals, RCKFAIL in RSTAT is set. The comparison is performed using unsigned arithmetic.
15-8	RMIN	0-FFh	Receive clock minimum boundary. This 8-bit unsigned value sets the minimum allowed boundary for the clock check counter after 32 receive high-frequency master clock (AHCLKR) signals have been received. If RCNT is less than RMIN after counting 32 AHCLKR signals, RCKFAIL in RSTAT is set. The comparison is performed using unsigned arithmetic.
7-4	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
3-0	RPS	0-Fh	Receive clock check prescaler value.
		0	McASP system clock divided by 1.
		1h	McASP system clock divided by 2.
		2h	McASP system clock divided by 4.
		3h	McASP system clock divided by 8.
		4h	McASP system clock divided by 16.
		5h	McASP system clock divided by 32.
		6h	McASP system clock divided by 64.
		7h	McASP system clock divided by 128.
		8h	McASP system clock divided by 256.
		9h-Fh	Reserved.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.23 Receiver DMA Event Control Register (REVTCTL)

The receiver DMA event control register (REVTCTL) contains a disable bit for the receiver DMA event. The REVTCTL is shown in Figure 21-46 and described in Table 21-33.

NOTE: Device-specific registers

Accessing REVTCTL not implemented on a specific device may cause improper device operation.

Figure 21-46. Receiver DMA Event Control Register (REVTCTL)

31		16
Reserved		
R-0		

15	1	0
Reserved		RDATDMA
R-0		R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 21-33. Receiver DMA Event Control Register (REVTCTL) Field Descriptions

Bit	Field	Value	Description
31-1	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
0	RDATDMA		Receive data DMA request enable bit. If writing to this bit, always write the default value of 0.
		0	Receive data DMA request is enabled.
		1	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.24 Transmitter Global Control Register (XGBLCTL)

Alias of the global control register (GBLCTL). Writing to the transmitter global control register (XGBLCTL) affects only the transmit bits of GBLCTL (bits 12-8). Reads from XGBLCTL return the value of GBLCTL. XGBLCTL allows the transmitter to be reset independently from the receiver. The XGBLCTL is shown in Figure 21-47 and described in Table 21-34. See Section 21.4.8 for a detailed description of GBLCTL.

Figure 21-47. Transmitter Global Control Register (XGBLCTL)

31						16
Reserved						
R-0						
15	13	12	11	10	9	8
Reserved		XFRST	XSMRST	XSRCLR	XHCLKRST	XCLKRST
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7	5	4	3	2	1	0
Reserved		RFRST	RSMRST	RSRCLR	RHCLKRST	RCLKRST
R-0		R-0	R-0	R-0	R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 21-34. Transmitter Global Control Register (XGBLCTL) Field Descriptions

Bit	Field	Value	Description
31-13	Reserved	0-FFh	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
12	XFRST		Transmit frame sync generator reset enable bit. A write to this bit affects the XFRST bit of GBLCTL.
		0	Transmit frame sync generator is reset.
		1	Transmit frame sync generator is active.
11	XSMRST		Transmit state machine reset enable bit. A write to this bit affects the XSMRST bit of GBLCTL.
		0	Transmit state machine is held in reset.
		1	Transmit state machine is released from reset.
10	XSRCLR		Transmit serializer clear enable bit. A write to this bit affects the XSRCLR bit of GBLCTL.
		0	Transmit serializers are cleared.
		1	Transmit serializers are active.
9	XHCLKRST		Transmit high-frequency clock divider reset enable bit. A write to this bit affects the XHCLKRST bit of GBLCTL.
		0	Transmit high-frequency clock divider is held in reset.
		1	Transmit high-frequency clock divider is running.
8	XCLKRST		Transmit clock divider reset enable bit. A write to this bit affects the XCLKRST bit of GBLCTL.
		0	Transmit clock divider is held in reset.
		1	Transmit clock divider is running.
7-5	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
4	RFRST	x	Receive frame sync generator reset enable bit. A read of this bit returns the RFRST bit value of GBLCTL. Writes have no effect.
3	RSMRST	x	Receive state machine reset enable bit. A read of this bit returns the RSMRST bit value of GBLCTL. Writes have no effect.
2	RSRCLR	x	Receive serializer clear enable bit. A read of this bit returns the RSRSCLR bit value of GBLCTL. Writes have no effect.
1	RHCLKRST	x	Receive high-frequency clock divider reset enable bit. A read of this bit returns the RHCLKRST bit value of GBLCTL. Writes have no effect.
0	RCLKRST	x	Receive clock divider reset enable bit. A read of this bit returns the RCLKRST bit value of GBLCTL. Writes have no effect.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.25 Transmit Format Unit Bit Mask Register (XMASK)

The transmit format unit bit mask register (XMASK) determines which bits of the transmitted data are masked off and padded with a known value before being shifted out the McASP. The XMASK is shown in Figure 21-48 and described in Table 21-35.

Figure 21-48. Transmit Format Unit Bit Mask Register (XMASK)

31	0
XMASK[*n*]	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 21-35. Transmit Format Unit Bit Mask Register (XMASK) Field Descriptions

Bit	Field	Value	Description
31-0	XMASK[31-0]		Transmit data mask *n* enable bit.
		0	Corresponding bit of transmit data (before passing through reverse and rotate units) is masked out and then padded with the selected bit pad value (XPAD and XPBIT bits in XFMT), which is transmitted out the McASP in place of the original bit.
		1	Corresponding bit of transmit data (before passing through reverse and rotate units) is transmitted out the McASP.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.26 Transmit Bit Stream Format Register (XFMT)

The transmit bit stream format register (XFMT) configures the transmit data format. The XFMT is shown in Figure 21-49 and described in Table 21-36.

Figure 21-49. Transmit Bit Stream Format Register (XFMT)

31–18	17–16
Reserved	XDATDLY
R-0	R/W-0

15	14–13	12–8	7–4	3	2–0
XRVRS	XPAD	XPBIT	XSSZ	XBUSEL	XROT
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 21-36. Transmit Bit Stream Format Register (XFMT) Field Descriptions

Bit	Field	Value	Description
31-18	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
17-16	XDATDLY	0-3h	Transmit sync bit delay.
		0	0-bit delay. The first transmit data bit, AXRn, occurs in same ACLKX cycle as the transmit frame sync (AFSX).
		1h	1-bit delay. The first transmit data bit, AXRn, occurs one ACLKX cycle after the transmit frame sync (AFSX).
		2h	2-bit delay. The first transmit data bit, AXRn, occurs two ACLKX cycles after the transmit frame sync (AFSX).
		3h	Reserved.
15	XRVRS		Transmit serial bitstream order.
		0	Bitstream is LSB first. No bit reversal is performed in transmit format bit reverse unit.
		1	Bitstream is MSB first. Bit reversal is performed in transmit format bit reverse unit.
14-13	XPAD	0-3h	Pad value for extra bits in slot not belonging to word defined by XMASK. This field only applies to bits when XMASK[n] = 0.
		0	Pad extra bits with 0.
		1h	Pad extra bits with 1.
		2h	Pad extra bits with one of the bits from the word as specified by XPBIT bits.
		3h	Reserved.
12-8	XPBIT	0-1Fh	XPBIT value determines which bit (as written by the CPU or DMA to XBUF[n]) is used to pad the extra bits before shifting. This field only applies when XPAD = 2h.
		0	Pad with bit 0 value.
		1-1Fh	Pad with bit 1 to bit 31 value.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 21-36. Transmit Bit Stream Format Register (XFMT) Field Descriptions (continued)

Bit	Field	Value	Description
7-4	XSSZ	0-Fh	Transmit slot size.
		0-2h	Reserved.
		3h	Slot size is 8 bits.
		4h	Reserved.
		5h	Slot size is 12 bits.
		6h	Reserved.
		7h	Slot size is 16 bits.
		8h	Reserved.
		9h	Slot size is 20 bits.
		Ah	Reserved.
		Bh	Slot size is 24 bits.
		Ch	Reserved.
		Dh	Slot size is 28 bits.
		Eh	Reserved.
		Fh	Slot size is 32 bits.
3	XBUSEL		Selects whether writes to serializer buffer XRBUF[n] originate from the configuration bus (CFG) or the data (DAT) port.
		0	Writes to XRBUF[n] originate from the data port. Writes to XRBUF[n] from the configuration bus are ignored with no effect to the McASP.
		1	Writes to XRBUF[n] originate from the configuration bus. Writes to XRBUF[n] from the data port are ignored with no effect to the McASP.
2-0	XROT	0-7h	Right-rotation value for transmit rotate right format unit.
		0	Rotate right by 0 (no rotation).
		1h	Rotate right by 4 bit positions.
		2h	Rotate right by 8 bit positions.
		3h	Rotate right by 12 bit positions.
		4h	Rotate right by 16 bit positions.
		5h	Rotate right by 20 bit positions.
		6h	Rotate right by 24 bit positions.
		7h	Rotate right by 28 bit positions.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.27 Transmit Frame Sync Control Register (AFSXCTL)

The transmit frame sync control register (AFSXCTL) configures the transmit frame sync (AFSX). The AFSXCTL is shown in Figure 21-50 and described in Table 21-37.

Figure 21-50. Transmit Frame Sync Control Register (AFSXCTL)

31							16
Reserved							
R-0							

15	7	6	5	4	3	2	1	0
XMOD		Reserved		FXWID	Reserved		FSXM	FSXP
R/W-0		R-0		R/W-0	R-0		R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 21-37. Transmit Frame Sync Control Register (AFSXCTL) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
15-7	XMOD	0-1FFh	Transmit frame sync mode select bits.
		0	Burst mode.
		1h	Reserved.
		2h-20h	2-slot TDM (I2S mode) to 32-slot TDM.
		21h-17Fh	Reserved.
		180h	384-slot DIT mode.
		181h-1FFh	Reserved.
6-5	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
4	FXWID		Transmit frame sync width select bit indicates the width of the transmit frame sync (AFSX) during its active period.
		0	Single bit.
		1	Single word.
3-2	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
1	FSXM		Transmit frame sync generation select bit.
		0	Externally-generated transmit frame sync.
		1	Internally-generated transmit frame sync.
0	FSXP		Transmit frame sync polarity select bit.
		0	A rising edge on transmit frame sync (AFSX) indicates the beginning of a frame.
		1	A falling edge on transmit frame sync (AFSX) indicates the beginning of a frame.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.28 Transmit Clock Control Register (ACLKXCTL)

The transmit clock control register (ACLKXCTL) configures the transmit bit clock (ACLKX) and the transmit clock generator. The ACLKXCTL is shown in Figure 21-51 and described in Table 21-38.

Figure 21-51. Transmit Clock Control Register (ACLKXCTL)

31					16
Reserved					
R-0					

15	8	7	6	5	4	0
Reserved		CLKXP	ASYNC	CLKXM	CLKXDIV	
R-0		R/W-0	R/W-1	R/W-1	R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 21-38. Transmit Clock Control Register (ACLKXCTL) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
7	CLKXP		Transmit bitstream clock polarity select bit.
		0	Rising edge. External receiver samples data on the falling edge of the serial clock, so the transmitter must shift data out on the rising edge of the serial clock.
		1	Falling edge. External receiver samples data on the rising edge of the serial clock, so the transmitter must shift data out on the falling edge of the serial clock.
6	ASYNC		Transmit/receive operation asynchronous enable bit.
		0	Synchronous. Transmit clock and frame sync provides the source for both the transmit and receive sections.
		1	Asynchronous. Separate clock and frame sync used by transmit and receive sections.
5	CLKXM		Transmit bit clock source bit.
		0	External transmit clock source from ACLKX pin.
		1	Internal transmit clock source from output of programmable bit clock divider.
4-0	CLKXDIV	0-1Fh	Transmit bit clock divide ratio bits determine the divide-down ratio from AHCLKX to ACLKX.
		0	Divide-by-1.
		1h	Divide-by-2.
		2h-1Fh	Divide-by-3 to divide-by-32.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.29 Transmit High-Frequency Clock Control Register (AHCLKXCTL)

The transmit high-frequency clock control register (AHCLKXCTL) configures the transmit high-frequency master clock (AHCLKX) and the transmit clock generator. The AHCLKXCTL is shown in Figure 21-52 and described in Table 21-39.

Figure 21-52. Transmit High-Frequency Clock Control Register (AHCLKXCTL)

31					16
Reserved					
R-0					

15	14	13	12	11	0
HCLKXM	HCLKXP	Reserved		HCLKXDIV	
R/W-1	R/W-0	R/W-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 21-39. Transmit High-Frequency Clock Control Register (AHCLKXCTL) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
15	HCLKXM		Transmit high-frequency clock source bit.
		0	External transmit high-frequency clock source from AHCLKX pin.
		1	Internal transmit high-frequency clock source from output of programmable high clock divider.
14	HCLKXP		Transmit bitstream high-frequency clock polarity select bit.
		0	AHCLKX is not inverted before programmable bit clock divider. In the special case where the transmit bit clock (ACLKX) is internally generated and the programmable bit clock divider is set to divide-by-1 (CLKXDIV = 0 in ACLKXCTL), AHCLKX is directly passed through to the ACLKX pin.
		1	AHCLKX is inverted before programmable bit clock divider. In the special case where the transmit bit clock (ACLKX) is internally generated and the programmable bit clock divider is set to divide-by-1 (CLKXDIV = 0 in ACLKXCTL), AHCLKX is directly passed through to the ACLKX pin.
13-12	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
11-0	HCLKXDIV	0-FFFh	Transmit high-frequency clock divide ratio bits determine the divide-down ratio from AUXCLK to AHCLKX.
		0	Divide-by-1.
		1h	Divide-by-2.
		2h-FFFh	Divide-by-3 to divide-by-4096.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.30 Transmit TDM Time Slot Register (XTDM)

The transmit TDM time slot register (XTDM) specifies in which TDM time slot the transmitter is active. TDM time slot counter range is extended to 384 slots (to support SPDIF blocks of 384 subframes). XTDM operates modulo 32, that is, XTDMS specifies the TDM activity for time slots 0, 32, 64, 96, 128, etc. The XTDM is shown in Figure 21-53 and described in Table 21-40.

Figure 21-53. Transmit TDM Time Slot Register (XTDM)

31	0
XTDMS[*n*]	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 21-40. Transmit TDM Time Slot Register (XTDM) Field Descriptions

Bit	Field	Value	Description
31-0	XTDMS[31-0]		Transmitter mode during TDM time slot *n*.
		0	Transmit TDM time slot *n* is inactive. The transmit serializer does not shift out data during this slot.
		1	Transmit TDM time slot *n* is active. The transmit serializer shifts out data during this slot according to the serializer control register (SRCTL).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.31 Transmitter Interrupt Control Register (XINTCTL)

The transmitter interrupt control register (XINTCTL) controls generation of the McASP transmit interrupt (XINT). When the register bit(s) is set to 1, the occurrence of the enabled McASP condition(s) generates XINT. The XINTCTL is shown in Figure 21-54 and described in Table 21-41. See Section 21.4.32 for a description of the interrupt conditions.

Figure 21-54. Transmitter Interrupt Control Register (XINTCTL)

31							8
Reserved							
R-0							
7	6	5	4	3	2	1	0
XSTAFRM	Reserved	XDATA	XLAST	XDMAERR	XCKFAIL	XSYNCERR	XUNDRN
R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 21-41. Transmitter Interrupt Control Register (XINTCTL) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
7	XSTAFRM		Transmit start of frame interrupt enable bit.
		0	Interrupt is disabled. A transmit start of frame interrupt does not generate a McASP transmit interrupt (XINT).
		1	Interrupt is enabled. A transmit start of frame interrupt generates a McASP transmit interrupt (XINT).
6	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
5	XDATA		Transmit data ready interrupt enable bit.
		0	Interrupt is disabled. A transmit data ready interrupt does not generate a McASP transmit interrupt (XINT).
		1	Interrupt is enabled. A transmit data ready interrupt generates a McASP transmit interrupt (XINT).
4	XLAST		Transmit last slot interrupt enable bit.
		0	Interrupt is disabled. A transmit last slot interrupt does not generate a McASP transmit interrupt (XINT).
		1	Interrupt is enabled. A transmit last slot interrupt generates a McASP transmit interrupt (XINT).
3	XDMAERR		Transmit DMA error interrupt enable bit.
		0	Interrupt is disabled. A transmit DMA error interrupt does not generate a McASP transmit interrupt (XINT).
		1	Interrupt is enabled. A transmit DMA error interrupt generates a McASP transmit interrupt (XINT).
2	XCKFAIL		Transmit clock failure interrupt enable bit.
		0	Interrupt is disabled. A transmit clock failure interrupt does not generate a McASP transmit interrupt (XINT).
		1	Interrupt is enabled. A transmit clock failure interrupt generates a McASP transmit interrupt (XINT).
1	XSYNCERR		Unexpected transmit frame sync interrupt enable bit.
		0	Interrupt is disabled. An unexpected transmit frame sync interrupt does not generate a McASP transmit interrupt (XINT).
		1	Interrupt is enabled. An unexpected transmit frame sync interrupt generates a McASP transmit interrupt (XINT).
0	XUNDRN		Transmitter underrun interrupt enable bit.
		0	Interrupt is disabled. A transmitter underrun interrupt does not generate a McASP transmit interrupt (XINT).
		1	Interrupt is enabled. A transmitter underrun interrupt generates a McASP transmit interrupt (XINT).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.32 Transmitter Status Register (XSTAT)

The transmitter status register (XSTAT) provides the transmitter status and transmit TDM time slot number. If the McASP logic attempts to set an interrupt flag in the same cycle that the CPU writes to the flag to clear it, the McASP logic has priority and the flag remains set. This also causes a new interrupt request to be generated. The XSTAT is shown in Figure 21-55 and described in Table 21-42.

Figure 21-55. Transmitter Status Register (XSTAT)

31						9	8
Reserved							XERR
R-0							R/W-0

7	6	5	4	3	2	1	0
XDMAERR	XSTAFRM	XDATA	XLAST	XTDMSLOT	XCKFAIL	XSYNCERR	XUNDRN
R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R-0	R/W1C-0	R/W1C-0	R/W1C-0

LEGEND: R/W = Read/Write; R = Read only; W1C = Writing a 1 clears this bit; writing a 0 has no effect; -*n* = value after reset

Table 21-42. Transmitter Status Register (XSTAT) Field Descriptions

Bit	Field	Value	Description
31-9	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
8	XERR		XERR bit always returns a logic-OR of:XUNDRN | XSYNCERR | XCKFAIL | XDMAERR Allows a single bit to be checked to determine if a transmitter error interrupt has occurred.
		0	No errors have occurred.
		1	An error has occurred.
7	XDMAERR		Transmit DMA error flag. XDMAERR is set when the CPU or DMA writes more serializers through the data port in a given time slot than were programmed as transmitters. Causes a transmit interrupt (XINT), if this bit is set and XDMAERR in XINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 has no effect.
		0	Transmit DMA error did not occur.
		1	Transmit DMA error did occur.
6	XSTAFRM		Transmit start of frame flag. Causes a transmit interrupt (XINT), if this bit is set and XSTAFRM in XINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 has no effect.
		0	No new transmit frame sync (AFSX) is detected.
		1	A new transmit frame sync (AFSX) is detected.
5	XDATA		Transmit data ready flag. Causes a transmit interrupt (XINT), if this bit is set and XDATA in XINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 has no effect.
		0	XBUF is written and is full.
		1	Data is copied from XBUF to XRSR. XBUF is empty and ready to be written. XDATA is also set when the transmit serializers are taken out of reset. When XDATA is set, it always causes a DMA event (AXEVT).
4	XLAST		Transmit last slot flag. XLAST is set along with XDATA, if the current slot is the last slot in a frame. Causes a transmit interrupt (XINT), if this bit is set and XLAST in XINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 has no effect.
		0	Current slot is not the last slot in a frame.
		1	Current slot is the last slot in a frame. XDATA is also set.
3	XTDMSLOT		Returns the LSB of XSLOT. Allows a single read of XSTAT to determine whether the current TDM time slot is even or odd.
		0	Current TDM time slot is odd.
		1	Current TDM time slot is even.
2	XCKFAIL		Transmit clock failure flag. XCKFAIL is set when the transmit clock failure detection circuit reports an error (see *Clock Failure Detection*). Causes a transmit interrupt (XINT), if this bit is set and XCKFAIL in XINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 has no effect.
		0	Transmit clock failure did not occur.
		1	Transmit clock failure did occur.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 21-42. Transmitter Status Register (XSTAT) Field Descriptions (continued)

Bit	Field	Value	Description
1	XSYNCERR		Unexpected transmit frame sync flag. XSYNCERR is set when a new transmit frame sync (AFSX) occurs before it is expected. Causes a transmit interrupt (XINT), if this bit is set and XSYNCERR in XINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 has no effect.
		0	Unexpected transmit frame sync did not occur.
		1	Unexpected transmit frame sync did occur.
0	XUNDRN		Transmitter underrun flag. XUNDRN is set when the transmit serializer is instructed to transfer data from XBUF to XRSR, but XBUF has not yet been serviced with new data since the last transfer. Causes a transmit interrupt (XINT), if this bit is set and XUNDRN in XINTCTL is set. This bit is cleared by writing a 1 to this bit. Writing a 0 has no effect.
		0	Transmitter underrun did not occur.
		1	Transmitter underrun did occur. For details on McASP action upon underrun conditions, see *Buffer Underrun Error - Transmitter*.

21.4.33 Current Transmit TDM Time Slot Register (XSLOT)

The current transmit TDM time slot register (XSLOT) indicates the current time slot for the transmit data frame. The XSLOT is shown in Figure 21-56 and described in Table 21-43.

Figure 21-56. Current Transmit TDM Time Slot Register (XSLOT)

31–16	15–10	9–0
Reserved	Reserved	XSLOTCNT
R-0	R-0	R-17Fh

LEGEND: R = Read only; -*n* = value after reset

Table 21-43. Current Transmit TDM Time Slot Register (XSLOT) Field Descriptions

Bit	Field	Value	Description
31-10	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
9-0	XSLOTCNT	0-17Fh	Current transmit time slot count. Legal values: 0 to 383 (17Fh). During reset, this counter value is 383 so the next count value, which is used to encode the first DIT group of data, will be 0 and encodes the B preamble. TDM function is not supported for >32 time slots. However, TDM time slot counter may count to 383 when used to transmit a DIT block.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.34 Transmit Clock Check Control Register (XCLKCHK)

The transmit clock check control register (XCLKCHK) configures the transmit clock failure detection circuit. The XCLKCHK is shown in Figure 21-57 and described in Table 21-44.

Figure 21-57. Transmit Clock Check Control Register (XCLKCHK)

31–24	23–16
XCNT	XMAX
R-0	R/W-0

15–8	7–4	3–0
XMIN	Reserved	XPS
R/W-0	R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 21-44. Transmit Clock Check Control Register (XCLKCHK) Field Descriptions

Bit	Field	Value	Description
31-24	XCNT	0	Transmit clock count value (from previous measurement). The clock circuit continually counts the number of system clocks for every 32 transmit high-frequency master clock (AHCLKX) signals, and stores the count in XCNT until the next measurement is taken.
23-16	XMAX	0-FFh	Transmit clock maximum boundary. This 8-bit unsigned value sets the maximum allowed boundary for the clock check counter after 32 transmit high-frequency master clock (AHCLKX) signals have been received. If the current counter value is greater than XMAX after counting 32 AHCLKX signals, XCKFAIL in XSTAT is set. The comparison is performed using unsigned arithmetic.
15-8	XMIN	0-FFh	Transmit clock minimum boundary. This 8-bit unsigned value sets the minimum allowed boundary for the clock check counter after 32 transmit high-frequency master clock (AHCLKX) signals have been received. If XCNT is less than XMIN after counting 32 AHCLKX signals, XCKFAIL in XSTAT is set. The comparison is performed using unsigned arithmetic.
7-4	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
3-0	XPS	0-Fh	Transmit clock check prescaler value.
		0	McASP system clock divided by 1.
		1h	McASP system clock divided by 2.
		2h	McASP system clock divided by 4.
		3h	McASP system clock divided by 8.
		4h	McASP system clock divided by 16.
		5h	McASP system clock divided by 32.
		6h	McASP system clock divided by 64.
		7h	McASP system clock divided by 128.
		8h	McASP system clock divided by 256.
		9h-Fh	Reserved.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.35 Transmitter DMA Event Control Register (XEVTCTL)

The transmitter DMA event control register (XEVTCTL) contains a disable bit for the transmit DMA event. The XEVTCTL is shown in Figure 21-58 and described in Table 21-45.

NOTE: Device-specific registers

Accessing REVTCTL not implemented on a specific device may cause improper device operation.

Figure 21-58. Transmitter DMA Event Control Register (XEVTCTL)

31		16
Reserved		
R-0		

15	1	0
Reserved		XDATDMA
R-0		R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 21-45. Transmitter DMA Event Control Register (XEVTCTL) Field Descriptions

Bit	Field	Value	Description
31-1	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
0	XDATDMA		Transmit data DMA request enable bit. If writing to this bit, always write the default value of 0.
		0	Transmit data DMA request is enabled.
		1	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.36 Serializer Control Registers (SRCTL*n*)

Each serializer on the McASP has a serializer control register (SRCTL). There are up to 6 serializers per McASP. The SRCTL is shown in Figure 21-59 and described in Table 21-46.

NOTE: Device-specific registers

Accessing SRCTL*n* not implemented on a specific device may cause improper device operation.

Figure 21-59. Serializer Control Registers (SRCTL*n*)

31–16	15–6	5	4	3–2	1–0
Reserved	Reserved	RRDY	XRDY	DISMOD	SRMOD
R-0	R-0	R-0	R-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 21-46. Serializer Control Registers (SRCTL*n*) Field Descriptions

Bit	Field	Value	Description
31-6	Reserved	0	Reserved. The reserved bit location always returns the default value. A value written to this field has no effect. If writing to this field, always write the default value for future device compatibility.
5	RRDY		Receive buffer ready bit. RRDY indicates the current receive buffer state. Always reads 0 when programmed as a transmitter or as inactive. If SRMOD bit is set to receive (2h), RRDY switches from 0 to 1 whenever data is transferred from XRSR to RBUF.
		0	Receive buffer (RBUF) is empty.
		1	Receive buffer (RBUF) contains data and needs to be read before the start of the next time slot or a receiver overrun occurs.
4	XRDY		Transmit buffer ready bit. XRDY indicates the current transmit buffer state. Always reads 0 when programmed as a receiver or as inactive. If SRMOD bit is set to transmit (1h), XRDY switches from 0 to 1 when XSRCLR in GBLCTL is switched from 0 to 1 to indicate an empty transmitter. XRDY remains set until XSRCLR is forced to 0, data is written to the corresponding transmit buffer, or SRMOD bit is changed to receive (2h) or inactive (0).
		0	Transmit buffer (XBUF) contains data.
		1	Transmit buffer (XBUF) is empty and needs to be written before the start of the next time slot or a transmit underrun occurs.
3-2	DISMOD	0-3h	Serializer pin drive mode bit. Drive on pin when in inactive TDM slot of transmit mode or when serializer is inactive. This field only applies if the pin is configured as a McASP pin (PFUNC = 0).
		0	Drive on pin is 3-state.
		1h	Reserved.
		2h	Drive on pin is logic low.
		3h	Drive on pin is logic high.
1-0	SRMOD	0-3h	Serializer mode bit.
		0	Serializer is inactive.
		1h	Serializer is transmitter.
		2h	Serializer is receiver.
		3h	Reserved.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.37 DIT Left Channel Status Registers (DITCSRA0-DITCSRA5)

The DIT left channel status registers (DITCSRA) provide the status of each left channel (even TDM time slot). Each of the six 32-bit registers (Figure 21-60) can store 192 bits of channel status data for a complete block of transmission. The DIT reuses the same data for the next block. It is your responsibility to update the register file in time, if a different set of data need to be sent.

Figure 21-60. DIT Left Channel Status Registers (DITCSRA0-DITCSRA5)

31	0
DITCSRA[*n*]	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

21.4.38 DIT Right Channel Status Registers (DITCSRB0-DITCSRB5)

The DIT right channel status registers (DITCSRB) provide the status of each right channel (odd TDM time slot). Each of the six 32-bit registers (Figure 21-61) can store 192 bits of channel status data for a complete block of transmission. The DIT reuses the same data for the next block. It is your responsibility to update the register file in time, if a different set of data need to be sent.

Figure 21-61. DIT Right Channel Status Registers (DITCSRB0-DITCSRB5)

31	0
DITCSRB[*n*]	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

21.4.39 DIT Left Channel User Data Registers (DITUDRA0-DITUDRA5)

The DIT left channel user data registers (DITUDRA) provides the user data of each left channel (even TDM time slot). Each of the six 32-bit registers (Figure 21-62) can store 192 bits of user data for a complete block of transmission. The DIT reuses the same data for the next block. It is your responsibility to update the register in time, if a different set of data need to be sent.

Figure 21-62. DIT Left Channel User Data Registers (DITUDRA0-DITUDRA5)

31	0
DITUDRA[*n*]	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

21.4.40 DIT Right Channel User Data Registers (DITUDRB0-DITUDRB5)

The DIT right channel user data registers (DITUDRB) provides the user data of each right channel (odd TDM time slot). Each of the six 32-bit registers (Figure 21-63) can store 192 bits of user data for a complete block of transmission. The DIT reuses the same data for the next block. It is your responsibility to update the register in time, if a different set of data need to be sent.

Figure 21-63. DIT Right Channel User Data Registers (DITUDRB0-DITUDRB5)

31	0
DITUDRB[*n*]	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.41 Transmit Buffer Registers (XBUF*n*)

The transmit buffers for the serializers (XBUF) hold data from the transmit format unit. For transmit operations, the XBUF (Figure 21-64) is an alias of the XRBUF in the serializer.

NOTE: Device-specific registers

Accessing XBUF registers not implemented on a specific device may cause improper device operation.

Figure 21-64. Transmit Buffer Registers (XBUFn)

31	0
XBUF[*n*]	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

21.4.42 Receive Buffer Registers (RBUF*n*)

The receive buffers for the serializers (RBUF) hold data from the serializer before the data goes to the receive format unit. For receive operations, the RBUF (Figure 21-65) is an alias of the XRBUF in the serializer.

NOTE: Device-specific registers

Accessing XBUF registers not implemented on a specific device may cause improper device operation.

Figure 21-65. Receive Buffer Registers (RBUF*n*)

31	0
RBUF[*n*]	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.43 Write FIFO Control Register (WFIFOCTL)

The Write FIFO control register (WFIFOCTL) is shown in Figure 21-66 and described in Table 21-47.

NOTE: The WNUMEVT and WNUMDMA values must be set prior to enabling the Write FIFO.

If the Write FIFO is to be enabled, it must be enabled prior to taking the McASP out of reset.

Figure 21-66. Write FIFO Control Register (WFIFOCTL)

31		17	16
Reserved			WENA
R-0			R/W-0

15	8	7	0
WNUMEVT		WNUMDMA	
R/W-10h		R/W-4h	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 21-47. Write FIFO Control Register (WFIFOCTL) Field Descriptions

Bit	Field	Value	Description
31-17	Reserved	0	Reserved
16	WENA		Write FIFO enable bit.
		0	Write FIFO is disabled. The WLVL bit in the Write FIFO status register (WFIFOSTS) is reset to 0 and pointers are initialized, that is, the Write FIFO is “flushed.”
		1	Write FIFO is enabled. If Write FIFO is to be enabled, it must be enabled prior to taking McASP out of reset.
15-8	WNUMEVT	0-FFh	Write word count per DMA event (32-bit). When the Write FIFO has space for at least WNUMEVT words of data, then an AXEVT (transmit DMA event) is generated to the host/DMA controller. This value should be set to a non-zero integer multiple of the number of serializers enabled as transmitters. This value must be set prior to enabling the Write FIFO.
		0	0 words
		1h	1 word
		2h	2 words
		3h-40h	3 to 64 words
		41h-FFh	Reserved
7-0	WNUMDMA	0-FFh	Write word count per transfer (32-bit words). Upon a transmit DMA event from the McASP, WNUMDMA words are transferred from the Write FIFO to the McASP. This value must equal the number of McASP serializers used as transmitters. This value must be set prior to enabling the Write FIFO.
		0	0 words
		1h	1 word
		2h	2 words
		3h-10h	3-16 words
		11h-FFh	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.44 Write FIFO Status Register (WFIFOSTS)

The Write FIFO status register (WFIFOSTS) is shown in Figure 21-67 and described in Table 21-48.

Figure 21-67. Write FIFO Status Register (WFIFOSTS)

31		16
Reserved		
R-0		

15	8	7	0
Reserved		WLVL	
R-0		R-0	

LEGEND: R = Read only; -*n* = value after reset

Table 21-48. Write FIFO Status Register (WFIFOSTS) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-0	WLVL	0-FFh	Write level (read-only). Number of 32-bit words currently in the Write FIFO.
		0	0 words currently in Write FIFO.
		1h	1 word currently in Write FIFO.
		2h	2 words currently in Write FIFO.
		3h-40h	3 to 64 words currently in Write FIFO.
		41h-FFh	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.45 Read FIFO Control Register (RFIFOCTL)

The Read FIFO control register (RFIFOCTL) is shown in Figure 21-68 and described in Table 21-49.

NOTE: The RNUMEVT and RNUMDMA values must be set prior to enabling the Read FIFO.

If the Read FIFO is to be enabled, it must be enabled prior to taking the McASP out of reset.

Figure 21-68. Read FIFO Control Register (RFIFOCTL)

31		17	16
Reserved			RENA
R-0			R/W-0

15	8	7	0
RNUMEVT		RNUMDMA	
R/W-10h		R/W-4h	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 21-49. Read FIFO Control Register (RFIFOCTL) Field Descriptions

Bit	Field	Value	Description
31-17	Reserved	0	Reserved
16	RENA		Read FIFO enable bit.
		0	Read FIFO is disabled. The RLVL bit in the Read FIFO status register (RFIFOSTS) is reset to 0 and pointers are initialized, that is, the Read FIFO is “flushed.”
		1	Read FIFO is enabled. If Read FIFO is to be enabled, it must be enabled prior to taking McASP out of reset.
15-8	RNUMEVT	0-FFh	Read word count per DMA event (32-bit). When the Read FIFO contains at least RNUMEVT words of data, then an AREVT (receive DMA event) is generated to the host/DMA controller. This value should be set to a non-zero integer multiple of the number of serializers enabled as receivers. This value must be set prior to enabling the Read FIFO.
		0	0 words
		1h	1 word
		2h	2 words
		3h-40h	3 to 64 words
		41h-FFh	Reserved
7-0	RNUMDMA	0-FFh	Read word count per transfer (32-bit words). Upon a receive DMA event from the McASP, the Read FIFO reads RNUMDMA words from the McASP. This value must equal the number of McASP serializers used as receivers. This value must be set prior to enabling the Read FIFO.
		0	0 words
		1	1 word
		2	2 words
		3h-10h	3-16 words
		11h-FFh	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

21.4.46 Read FIFO Status Register (RFIFOSTS)

The Read FIFO status register (RFIFOSTS) is shown in Figure 21-69 and described in Table 21-50.

Figure 21-69. Read FIFO Status Register (RFIFOSTS)

31	16
Reserved	
R-0	

15	8	7	0
Reserved		RLVL	
R-0		R-0	

LEGEND: R = Read only; -*n* = value after reset

Table 21-50. Read FIFO Status Register (RFIFOSTS) Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7-0	RLVL	0-FFh	Read level (read-only). Number of 32-bit words currently in the Read FIFO.
		0	0 words currently in Read FIFO.
		1h	1 word currently in Read FIFO.
		2h	2 words currently in Read FIFO.
		3h-40h	3 to 64 words currently in Read FIFO.
		41h-FFh	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Chapter 22

SPRUH73–July 2011

Controller Area Network (CAN)

This chapter describes the controller area network for the device.

Topic **Page**

22.1 Introduction **2774**
22.2 Integration **2775**
22.3 Functional Description **2776**
22.4 Use Cases **2813**
22.5 DCAN Control Registers **2814**

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.1 Introduction

22.1.1 DCAN Features

The general features of the DCAN controller are:

- Supports CAN protocol version 2.0 part A, B (ISO 11898-1)
- Bit rates up to 1 MBit/s
- Dual clock source
- 16, 32, 64 or 128 message objects (instantiated as 64 on this device)
- Individual identifier mask for each message object
- Programmable FIFO mode for message objects
- Programmable loop-back modes for self-test operation
- Suspend mode for debug support
- Software module reset
- Automatic bus on after Bus-Off state by a programmable 32-bit timer
- Message RAM parity check mechanism
- Direct access to Message RAM during test mode
- CAN Rx / Tx pins configurable as general purpose IO pins
- Two interrupt lines (plus additional parity-error interrupt line)
- RAM initialization
- DMA support
- OCP interface up to 100 MHz (32b slave interface)

22.1.2 Unsupported DCAN Features

The DCAN module in this device does not support GPIO pin mode. All GPIO functionality is mapped through the GPIO modules and muxed at the pins. GPIO pin control signals from the DCAN modules are not connected.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.2 Integration

The Controller Area Network is a serial communications protocol which efficiently supports distributed realtime control with a high level of security. The DCAN module supports bitrates up to 1 Mbit/s and is compliant to the CAN 2.0B protocol specification. The core IP within DCAN is provided by Bosch.

This device includes two instantiations of the DCAN controller: DCAN0 and DCAN1. Figure 22-1 shows the DCAN module integration.

Figure 22-1. DCAN Integration

22.2.1 DCAN Connectivity Attributes

The general connectivity attributes for the DCAN module are shown in Table 22-1.

Table 22-1. DCAN Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_L4LS_GCLK (OCP) PD_PER_CAN_CLK (Func)
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	3 Interrupts Intr0 – Error, Status, Msg Object interrupt Intr1 – Msg Object interrupt Uerr – Parity error interrupt All DCAN0 interrupts to MPU Subsystem and PRUSS. All DCAN1 interrupts to MPU Subsystem only.
DMA Requests	3 DMA requests to EDMA
Physical Address	L4 Peripheral slave port

22.2.2 DCAN Clock and Reset Management

The DCAN controllers have separate bus interface and functional clocks.

Table 22-2. DCAN Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
DCAN_ocp_clk Interface clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_l4ls_gclk from PRCM
DCAN_io_clk Functional clock	24 MHz	CLK_M_OSC	pd_per_can_clk from PRCM

22.2.3 DCAN Pin List

The external signals for the DCAN module are shown in the following table.

Table 22-3. DCAN Pin List

Pin	Type	Description
DCAN*x*_TX	O	DCAN transmit line
DCAN*x*_RX	I	DCAN receive line

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.3 Functional Description

The DCAN module performs CAN protocol communication according to ISO 11898-1. The bit rate can be programmed to values up to 1 MBit/s. Additional transceiver hardware is required for the connection to the physical layer (CAN bus).

For communication on a CAN network, individual message objects can be configured. The message objects and identifier masks are stored in the message RAM.

All functions concerning the handling of messages are implemented in the message handler. Those functions are acceptance filtering, the transfer of messages between the CAN core and the message RAM, and the handling of transmission requests, as well as the generation of interrupts or DMA requests.

The register set of the DCAN module can be accessed directly by the CPU via the module interface. These registers are used to control and configure the CAN core and the message handler, and to access the message RAM.

Figure 22-2 shows the DCAN block diagram and its features are described below.

Figure 22-2. DCAN Block Diagram

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.3.0.1 CAN Core

The CAN core consists of the CAN protocol controller and the Rx/Tx shift register. It handles all ISO 11898-1 protocol functions.

22.3.0.2 Message Handler

The message handler is a state machine that controls the data transfer between the single-ported message RAM and the CAN core's Rx/Tx shift register. It also handles acceptance filtering and the interrupt/DMA request generation as programmed in the control registers.

22.3.0.3 Message RAM

The DCAN0 and DCAN1 enables a storage of 64 CAN messages.

22.3.0.4 Message RAM Interface

Three interface register sets control the CPU read and write accesses to the message RAM. There are two interface registers sets for read and write access, IF1 and IF2, and one interface register set for read access only, IF3. Additional information can be found in Section 22.3.8.12.

The interface registers have the same word-length as the message RAM.

22.3.0.5 Registers and Message Object Access

Data consistency is ensured by indirect accesses to the message objects. During normal operation, all CPU and DMA accesses to the message RAM are done through interface registers. In a dedicated test mode, the message RAM is memory mapped and can be directly accessed by either CPU or DMA.

22.3.0.6 Module Interface

The DCAN module registers are accessed by the CPU or user software through a 32-bit peripheral bus interface.

22.3.0.7 Dual Clock Source

Two clock domains are provided to the DCAN module: the peripheral synchronous clock domain (L3_SLOW_GCLK) and the peripheral asynchronous clock source domain (OSC0) for CAN_CLK.

22.3.1 CAN Operation

After hardware reset, the Init bit in the CAN control register (DCAN CTL) is set and all CAN protocol functions are disabled. The CAN module must be initialized before operating it. Figure 22-3 illustrates the basic initialization flow for the CAN module.

22.3.1.1 CAN Module Initialization

A general CAN module initialization would mean the following two critical steps:

- Configuration of the CAN bit timing
- Configuration of message objects

To initialize the CAN controller, the CPU has to set up the CAN bit timing and those message objects that have to be used for CAN communication. Message objects that are not needed, can be deactivated.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 22-3. CAN Module General Initialization Flow

22.3.1.1.1 Configuration of CAN Bit Timing

The CAN module must be in initialization mode to configure the CAN bit timing.

For CAN bit timing software configuration flow, see Figure 22-4.

Step 1: Enter *initialization mode* by setting the Init (Initialization) bit in the CAN control register.

While the Init bit is set, the message transfer from and to the CAN bus is stopped, and the status of the CAN_TX output is recessive (high).

The CAN error counters are not updated. Setting the Init bit does not change any other configuration register.

Also, note that the CAN module is in initialization mode on hardware reset and during Bus-Off.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 22-4. CAN Bit-Timing Configuration

Step 2: Set the Configure Change Enable (CCE) bit in the CAN control register.

The access to the Bit Timing register (BTR) for the configuration of the bit timing is enabled when both Init and CCE bits in the CAN Control register are set.

Step 3: Wait for the Init bit to get set. This would make sure that the module has entered *Initialization mode*.

Step 4: Write the bit timing values into the bit timing register. See Section 22.3.9.2 for the BTR value calculation for a given bit timing.

Step 5: Clear the CCE and Init bit.

Step 6: Wait for the Init bit to clear. This would ensure that the module has come out of *initialization mode*.

Following these steps, the module comes to operation by synchronizing itself to the CAN bus, provided the BTR is configured as per the CAN bus baud rate, although the message objects have to be configured before carrying out any communication.

NOTE: The module will not come out of the *initialization mode* if any incorrect BTR values are written in step 4.

NOTE: The required message objects should be configured as transmit or receive objects before the start of data transfer as explained in Section 22.3.1.1.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.3.1.1.2 Configuration of Message Objects

The message objects can be configured only through the interface registers; the CPU does not have direct access to the message object (message RAM) . Familiarize yourself with the interface register set (IFx) usage (see Section 22.3.10) and the message object structure (see Section 22.3.11) before configuring the message objects.

For more information regarding the procedure to configure the message objects, see Section 22.3.7. All the message objects should be configured to particular identifiers or set to not valid before the message transfer is started. It is possible to change the configuration of message objects during normal operation (that is between data transfers).

NOTE: The message objects initialization is independent of the bit-timing configuration.

22.3.1.1.3 DCAN RAM Hardware Initialization

The memory hardware initialization for the DCAN module is enabled in the device control register, DCAN_RAMINIT, which initializes the RAM with zeros and sets parity bits accordingly. Wait for the RAMINIT_DONE bit to be set to ensure successful RAM initialization.

For more details on RAM hardware initialization, see the system module reference guide.

22.3.1.2 CAN Message Transfer (Normal Operation)

Once the DCAN is initialized and Init bit is reset to zero, the CAN core synchronizes itself to the CAN bus and is ready for message transfer as per the configured message objects.

The CPU may enable the interrupt lines (setting IE0 and IE1 to ‘1’) at the same time when it clears Init and CCE. The status interrupts EIE and SIE may be enabled simultaneously.

The CAN communication can be carried out in any of the following two modes: interrupt and polling.

The interrupt register points to those message objects with IntPnd = ‘1’. It is updated even if the interrupt lines to the CPU are disabled (IE0/IE1 are zero).

The CPU may poll all MessageObject’s NewDat and TxRqst bits in parallel from the NewData X registers and the Transmission Request X Registers (DCAN TXRQ X). Polling can be made easier if all transmit objects are grouped at the low numbers and all receive objects are grouped at the high numbers.

Received messages are stored into their appropriate message objects if they pass acceptance filtering.

The whole message (including all arbitration bits, DLC and up to eight data bytes) is stored into the message object. As a consequence (e.g., when the identifier mask is used), the arbitration bits which are masked to “don’t care” may change in the message object when a received message is stored.

The CPU may read or write each message at any time via the interface registers, as the message handler guarantees data consistency in case of concurrent accesses.

If a permanent message object (arbitration and control bits set up during configuration and leaving unchanged for multiple CAN transfers) exists for the message, it is possible to only update the data bytes.

If several transmit messages should be assigned to one message object, the whole message object has to be configured before the transmission of this message is requested.

The transmission of multiple message objects may be requested at the same time. They are subsequently transmitted, according to their internal priority.

Messages may be updated or set to not valid at any time, even if a requested transmission is still pending. However, the data bytes will be discarded if a message is updated before a pending transmission has started.

Depending on the configuration of the message object, a transmission may be automatically requested by the reception of a remote frame with a matching identifier.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.3.1.2.1 Automatic Retransmission

According to the CAN Specification (ISO11898), the DCAN provides a mechanism to automatically retransmit frames which have lost arbitration or have been disturbed by errors during transmission. The frame transmission service will not be confirmed to the user before the transmission is successfully completed.

By default, this automatic retransmission is enabled. It can be disabled by setting bit disable automatic retransmission (DAR) in the CAN Control register. Further details to this mode are provided in Section 22.3.8.3.

22.3.1.2.2 Auto-Bus-On

By default, after the DCAN has entered Bus-Off state, the CPU can start a Bus-Off-Recovery sequence by resetting the Init bit. If this is not done, the module will stay in Bus-Off state.

The DCAN provides an automatic Auto-Bus-On feature which is enabled by bit ABO in the CAN control register. If set, the DCAN will automatically start the Bus-Off-Recovery sequence. The sequence can be delayed by a user-defined number of L3_SLOW_GCLK cycles which can be defined in the Auto-Bus-On Time register (DCAN ABOTR).

NOTE: If the DCAN goes to Bus-Off state due to a massive occurrence of CAN bus errors, it stops all bus activities and automatically sets the Init bit. Once the Init bit has been reset by the CPU or due to the Auto-Bus-On feature, the device will wait for 129 occurrences of bus Idle (equal to 129 * 11 consecutive recessive bits) before resuming normal operation. At the end of the Bus-Off recovery sequence, the error counters will be reset.

22.3.1.3 Test Modes

The DCAN module provides several test modes which are mainly intended for production tests or self test.

For all test modes, the Test bit in the CAN control register needs to be set to one. This enables write access to the test register (DCAN TEST).

22.3.1.3.1 Silent Mode

The silent mode may be used to analyze the traffic on the CAN bus without affecting it by sending dominant bits (e.g., acknowledge bit, overload flag, active error flag). The DCAN is still able to receive valid data frames and valid remote frames, but it will not send any dominant bits. However, these are internally routed to the CAN core.

Figure 22-5 shows the connection of signals CAN_TX and CAN_RX to the CAN core in silent mode. Silent mode can be activated by setting the Silent bit in the Test register to one. In ISO 11898-1, the silent mode is called the bus monitoring mode.

Figure 22-5. CAN Core in Silent Mode

Copyright © 2011, Texas Instruments Incorporated

22.3.1.3.2 Loopback Mode

The loopback mode is mainly intended for hardware self-test functions. In this mode, the CAN core uses internal feedback from Tx output to Rx input. Transmitted messages are treated as received messages, and can be stored into message objects if they pass acceptance filtering. The actual value of the CAN_RX input pin is disregarded by the CAN core. Transmitted messages can still be monitored at the CAN_TX pin.

In order to be independent from external stimulation, the CAN core ignores acknowledge sampled in the acknowledge slot of a data/remote frame.

Figure 22-6 shows the connection of signals CAN_TX and CAN_RX to the CAN core in loopback mode.

Loopback mode can be activated by setting bit LBack in the test register to one.

NOTE: In loopback mode, the signal path from CAN core to Tx pin, the Tx pin itself, and the signal path from Tx pin back to CAN core are disregarded. For including these into the testing, see Section 22.3.1.3.3.

Figure 22-6. CAN Core in Loopback Mode

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.3.1.3.3 External Loopback Mode

The external loopback mode is similar to the loopback mode; however, it includes the signal path from CAN core to Tx pin, the Tx pin itself, and the signal path from Tx pin back to CAN core. When external loopback mode is selected, the input of the CAN core is connected to the input buffer of the Tx pin.

With this configuration, the Tx pin IO circuit can be tested.

External loopback mode can be activated by setting bit **ExL** in test register to one.

Figure 22-7 shows the connection of signals CAN_TX and CAN_RX to the CAN core in external loopback mode.

NOTE: When loopback mode is active (LBack bit set), the ExL bit will be ignored.

Figure 22-7. CAN Core in External Loopback Mode

22.3.1.3.4 Loopback Mode Combined With Silent Mode

It is also possible to combine loopback mode and silent mode by setting bits LBack and Silent at the same time. This mode can be used for a “Hot Selftest,” i.e., the DCAN hardware can be tested without affecting the CAN network. In this mode, the CAN_RX pin is disconnected from the CAN core and no dominant bits will be sent on the CAN_TX pin.

Figure 22-8 shows the connection of the signals CAN_TX and CAN_RX to the CAN core in case of the combination of loopback mode with silent mode.

Copyright © 2011, Texas Instruments Incorporated

Figure 22-8. CAN Core in Loop Back Combined With Silent Mode

22.3.1.3.5 Software Control of CAN_TX pin

Four output functions are available for the CAN transmit pin CAN_TX. In addition to its default function (serial data output), the CAN_TX pin can drive constant dominant or recessive values, or it can drive the CAN sample point signal to monitor the CAN core's bit timing.

Combined with the readable value of the CAN_RX pin, this function can be used to check the physical layer of the CAN bus.

The output mode of pin CAN_TX is selected by programming the test register bits Tx[1:0] as described in section Section 22.5.6.

NOTE: The software control for the CAN_TX pin interferes with CAN protocol functions. For CAN message transfer or any of the test modes (loopback mode, external loopback mode or silent mode), the CAN_TX pin should operate in its default functionality.

22.3.2 Dual Clock Source

Two clock domains are provided to the DCAN module: the peripheral synchronous clock domain (L3_SLOW_GCLK) as the general module clock source, and the peripheral asynchronous clock source domain (OSC0) provided to the CAN core (as clock source CAN_CLK) for generating the CAN bit timing.

Both clock domains can be derived from the same clock source (so that L3_SLOW_GCLK = OSC0).

For more information on how to configure the relevant clock source registers in the system module, see the system module reference guide and the device-specific data sheet.

Between the two clock domains, a synchronization mechanism is implemented in the DCAN module in order to ensure correct data transfer.

NOTE: If the dual clock functionality is used, then L3_SLOW_GCLK must always be higher or equal to CAN_CLK (OSC0) (derived from the asynchronous clock source), in order to achieve a stable functionality of the DCAN. Here also the frequency shift of the modulated L3_SLOW_GCLK has to be considered:

$f_{0,\ L3_SLOW_GCLK}(OCP) \pm \Delta f_{FM,\ L3_SLOW_GCLK}(OCP) \geq f_{CANCLK}$

NOTE: The CAN core has to be programmed to at least 8 clock cycles per bit time. To achieve a transfer rate of 1 MBaud when using the asynchronous clock domain as the clock source for CAN_CLK (OSC0), an oscillator frequency of 8MHz or higher has to be used.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.3.3 Interrupt Functionality

Interrupts can be generated on two interrupt lines: DCANINT0 and DCANINT1. These lines can be enabled by setting the IE0 and IE1 bits, respectively, in the CAN control register. The interrupts are level triggered at the chip level.

The DCAN provides three groups of interrupt sources: message object interrupts, status change interrupts, and error interrupts (see Figure 22-9 and Figure 22-10).

The source of an interrupt can be determined by the interrupt identifiers Int0ID/Int1ID in the interrupt register (see Section 22.5.5). When no interrupt is pending, the register will hold the value zero.

Each interrupt line remains active until the dedicated field in the interrupt register DCAN INT (Int0ID / Int1ID) again reach zero (this means the cause of the interrupt is reset), or until IE0 / IE1 are reset.

The value 0x8000 in the Int0ID field indicates that an interrupt is pending because the CAN core has updated (not necessarily changed) the Error and Status register (DCAN ES and PARITYERR_EOI). This interrupt has the highest priority. The CPU can update (reset) the status bits WakeUpPnd, RxOk, TxOk and LEC by reading the error and status register DCAN ES, but a write access of the CPU will never generate or reset an interrupt.

Values between 1 and the number of the last message object indicates that the source of the interrupt is one of the message objects, Int0ID resp. Int1ID will point to the pending message interrupt with the highest priority. The Message Object 1 has the highest priority; the last message object has the lowest priority.

An interrupt service routine that reads the message that is the source of the interrupt may read the message and reset the message object's IntPnd at the same time (ClrIntPnd bit in the IF1/IF2 command register). When IntPnd is cleared, the interrupt register will point to the next message object with a pending interrupt.

22.3.3.1 Message Object Interrupts

Message object interrupts are generated by events from the message objects. They are controlled by the flags IntPND, TxIE and RxIE that are described in Section 22.3.11.1.

Message object interrupts can be routed to either DCANINT0 or DCANINT1 line, controlled by the interrupt multiplexer register (DCAN INTMUX12 to DCAN INTMUX78), see Section 22.5.17.

22.3.3.2 Status Change Interrupts

The events WakeUpPnd, RxOk, TxOk and LEC in error and status register (DCAN ES) belong to the status change interrupts. The status change interrupt group can be enabled by bit in CAN control register.

If SIE is set, a status change interrupt will be generated at each CAN frame, independent of bus errors or valid CAN communication, and also independent of the message RAM configuration.

Status change interrupts can only be routed to interrupt line DCAN0INT, which has to be enabled by setting IE0 in the CAN control register.

NOTE: Reading the error and status register will clear the WakeUpPnd flag. If in global power-down mode, the WakeUpPnd flag is cleared by such a read access before the DCAN module has been waken up by the system, the DCAN may re-assert the WakeUpPnd flag, and a second interrupt may occur.

22.3.3.3 Error Interrupts

The events PER, BOff and EWarn, monitored in the Error and Status register, DCAN ES, belong to the error interrupts. The error interrupt group can be enabled by setting bit EIE in the CAN Control register.

Error interrupts can only be routed to interrupt line DCAN0INT, which has to be enabled by setting IE0 in the CAN Control register.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 22-9. CAN Interrupt Topology 1

Figure 22-10. CAN Interrupt Topology 2

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.3.4 Local Power-Down Mode

The DCAN supports a local power-down mode, which can be controlled within the DCAN control registers.

22.3.4.1 Entering Local Power-Down Mode

The local power-down mode is requested by setting the PDR bit in CAN Control register.

The DCAN then finishes all transmit requests of the message objects. When all requests are done, DCAN waits until a bus idle state is recognized. Then it will automatically set the Init bit in CAN control register to prevent any further CAN transfers, and it will also set the PDA bit in CAN error and status register. With setting the PDA bits, the DCAN module indicates that the local power-down mode has been entered.

During local power-down mode, the internal clocks of the DCAN module are turned off, but there is a wakeup logic (see Section 22.3.4.2) that can be active, if enabled. Also, the actual contents of the control registers can be read back.

NOTE: In local low-power mode, the application should not clear the Init bit while PDR is set. If there are any messages in the message RAM which are configured as transmit messages and the application resets the init bit, these messages may get sent.

22.3.4.2 Wakeup From Local Power Down

There are two ways to wake up the DCAN from local power-down mode:

- The application could wake up the DCAN module manually by clearing the PDR bit and then clearing the Init bit in CAN Control register.
- Alternatively, a CAN bus activity detection circuit can be activated by setting the wakeup on bus activity bit (WUBA) in the CAN control register. If this circuit is active, on occurrence of a dominant CAN bus level, the DCAN will automatically start the wakeup sequence. It will clear the PDR bit in the CAN Control register and also clear the PDA bit in the error and status register. The WakeUpPnd bit in CAN error and status register will be set. If status interrupts are enabled, also an interrupt will be generated. Finally the Init bit in CAN control register will be cleared.

After the Init bit has been cleared, the module waits until it detects 11 consecutive recessive bits on the CAN_RX pin and then goes bus-active again.

NOTE: The CAN transceiver circuit has to stay active in order to detect any CAN bus activity while the DCAN is in local power down mode. The first CAN message, which initiates the bus activity, cannot be received. This means that the first message received in power-down and automatic wake-up mode, is lost.

Figure 22-11 shows a flow diagram about entering and leaving local power-down mode.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 22-11. Local Power-Down Mode Flow Diagram

Copyright © 2011, Texas Instruments Incorporated

22.3.5 Parity Check Mechanism

The DCAN provides a parity check mechanism to ensure data integrity of message RAM data. For each word (32 bits) in message RAM, one parity bit will be calculated. The formation of the different words is according to the message RAM representation in RDA mode, see Section 22.3.11.4.

Parity information is stored in the message RAM on write accesses and will be checked against the stored parity bit from message RAM on read accesses.

The Parity check functionality can be enabled or disabled by PMD bit field in the CAN control register.

In case of disabled parity check, the parity bits in message RAM will be left unchanged on write access to data area and no check will be done on read access.

If parity checking is enabled, parity bits will be automatically generated and checked by the DCAN. The parity bits could be read in debug/suspend mode (see Section 22.3.11.3) or in RDA mode (see Section 22.3.11.4). However, direct write access to the parity bits is only possible in this two modes with parity check disabled.

A parity bit will be set, if the modulo-2-sum of the data bits is 1. This definition is equivalent to: The parity bit will be set, if the number of 1 bits in the data is odd.

NOTE: The parity scheme is tied to even parity at the device level.

22.3.5.1 Behavior on Parity Error

On any read access to message RAM (e.g., during start of a CAN frame transmission), the parity of the message object will be checked. If a parity error is detected, the PER bit in the error and status register will be set. If error interrupts are enabled, an interrupt would also be generated. In order to avoid the transmission of invalid data over the CAN bus, the d bit of the message object will be reset.

The message object data can be read by the host CPU, independently of parity errors. Thus, the application has to ensure that the read data is valid, e.g., by immediately checking the parity error code register (DCAN PERR) on parity error interrupt.

NOTE: During RAM initialization, no parity check will be done.

22.3.5.2 Parity Testing

Testing the parity mechanism can be done by enabling the bit RamDirectAccess (RDA) and manually writing the parity bits directly to the dedicated RAM locations. With this, data and parity bits could be checked when reading directly from RAM.

NOTE: If parity check was disabled, the application has to ensure correct parity bit handling in order to prevent parity errors later on when parity check is enabled.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.3.6 Debug/Suspend Mode

The module supports the usage of an external debug unit by providing functions like pausing DCAN activities and making message RAM content accessible via OCP interface.

Before entering debug/suspend mode, the circuit will either wait until a started transmission or reception will be finished and bus idle state is recognized, or immediately interrupt a current transmission or reception. This is depending on bit IDS in the CAN control register.

Afterwards, the DCAN enters debug/suspend mode, indicated by InitDbg flag in the CAN control register.

During debug/suspend mode, all DCAN registers can be accessed. Reading reserved bits will return ‘0’. Writing to reserved bits will have no effect.

Also, the message RAM will be memory mapped. This allows the external debug unit to read the message RAM. For the memory organization, see Section 22.3.11.3).

NOTE: During debug/suspend mode, the message RAM cannot be accessed via the IFx register sets.

NOTE: Writing to control registers in debug/suspend mode may influence the CAN state machine and further message handling.

For debug support, the auto clear functionality of the following DCAN registers is disabled:

- Error and status register (clear of status flags by read)
- IF1/IF2 command registers (clear of DMAActive flag by read/write)

22.3.7 Configuration of Message Objects

The whole message RAM should be configured before the end of the initialization, however it is also possible to change the configuration of message objects during CAN communication.

The CAN software driver library should offer subroutines that:

- Transfer a complete message structure into a message object. (Configuration)
- Transfer the data bytes of a message into a message object and set TxRqst and NewDat. (Start a new transmission)
- Get the data bytes of a message from a message object and clear NewDat (and IntPnd). (Read received data)
- Get the complete message from a message object and clear NewDat (and IntPnd). (Read a received message, including identifier, from a message object with UMask = ‘1’)

Parameters of the subroutines are the Message Number and a pointer to a complete message structure or to the data bytes of a message structure.

Two examples of assigning the IFx interface register sets to these subroutines are shown here:

In the first method, the tasks of the application program that may access the module are divided into two groups. Each group is restricted to the use of one of the interface register sets. The tasks of one group may interrupt tasks of the other group, but not of the same group.

In the second method, which may be a special case of the first method, there are only two tasks in the application program that access the module. A Read_Message task that uses IF2 or IF3 to get received messages from the message RAM and a Write_Message task that uses IF1 to write messages to be transmitted (or to be configured) into the message RAM. Both tasks may interrupt each other.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.3.7.1 Configuration of a Transmit Object for Data Frames

Table 22-4 shows how a transmit object can be initialized.

Table 22-4. Initialization of a Transmit Object

MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	1	0	0	0	appl.	0	appl.	0

The arbitration bits (ID[28:0] and Xtd bit) are given by the application. They define the identifier and type of the outgoing message. If an 11-bit identifier (standard frame) is used (Xtd = ‘0’), it is programmed to ID[28:18]. In this case, ID[17:0] can be ignored.

The data registers (DLC[3:0] and Data0-7) are given by the application. TxRqst and RmtEn should not be set before the data is valid.

If the TxIE bit is set, the IntPnd bit will be set after a successful transmission of the message object.

If the RmtEn bit is set, a matching received remote frame will cause the TxRqst bit to be set; the remote frame will autonomously be answered by a data frame.

The mask bits (Msk[28:0], UMask, MXtd, and MDir bits) may be used (UMask=’1’) to allow groups of remote frames with similar identifiers to set the TxRqst bit. The Dir bit should not be masked. For details, see Section 22.3.8.8. Identifier masking must be disabled (UMask = ‘0’) if no remote frames are allowed to set the TxRqst bit (RmtEn = ‘0’).

22.3.7.2 Configuration of a Transmit Object for Remote Frames

It is not necessary to configure transmit objects for the transmission of remote frames. Setting TxRqst for a receive object causes the transmission of a remote frame with the same identifier as the data rrame for which this receive object is configured.

22.3.7.3 Configuration of a Single Receive Object for Data Frames

Table 22-5 shows how a receive object for data frames can be initialized.

Table 22-5. Initialization of a single Receive Object for Data Frames

MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	0	0	0	appl.	0	0	0	0

The arbitration bits (ID[28:0] and Xtd bit) are given by the application. They define the identifier and type of accepted received messages. If an 11-bit Identifier (standard frame) is used (Xtd = ‘0’), it is programmed to ID[28:18]. In this case, ID[17:0] can be ignored. When a data frame with an 11-bit Identifier is received, ID[17:0] is set to ‘0’.

The data length code (DLC[3:0]) is given by the application. When the message handler stores a data frame in the message object, it will store the received data length code and eight data bytes. If the data length code is less than 8, the remaining bytes of the message object may be overwritten by non specified values.

The mask bits (Msk[28:0], UMask, MXtd, and MDir bits) may be used (UMask = ’1’) to allow groups of data frames with similar identifiers to be accepted. The Dir bit should not be masked in typical applications. If some bits of the mask bits are set to “don’t care,” the corresponding bits of the arbitration register will be overwritten by the bits of the stored data frame.

If the RxIE bit is set, the IntPnd bit will be set when a received data frame is accepted and stored in the message object.

If the TxRqst bit is set, the transmission of a remote frame with the same identifier as actually stored in the arbitration bits will be triggered. The content of the arbitration bits may change if the mask bits are used (UMask = ’1’) for acceptance filtering.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.3.7.4 Configuration of a Single Receive Object for Remote Frames

Table 22-6 shows how a receive object for remote frames can be initialized.

Table 22-6. Initialization of a Single Receive Object for Remote Frames

MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	1	0	0	appl.	0	0	0	0

A receive object for remote frames may be used to monitor remote frames on the CAN bus. The remote frame stored in the receive object will not trigger the transmission of a data frame. Receive objects for remote frames may be expanded to a FIFO buffer (see Section 22.3.7.5).

UMask must be set to ‘1.’ The mask bits (Msk[28:0], UMask, MXtd, and MDir bits) may be set to “must-match” or to “don’t care,” to allow groups of remote frames with similar identifiers to be accepted. The Dir bit should not be masked in typical applications. For details, see Section 22.3.8.8.

The arbitration bits (ID[28:0] and Xtd bit) may be given by the application. They define the identifier and type of accepted received remote frames. If some bits of the mask bits are set to “don’t care,” the corresponding bits of the arbitration bits will be overwritten by the bits of the stored remote frame. If an 11-bit Identifier (standard frame) is used (Xtd = ‘0’), it is programmed to ID[28:18]. In this case, ID[17:0] can be ignored. When a remote frame with an 11-bit Identifier is received, ID[17:0] will be set to ‘0.’

The data length code (DLC[3:0]) may be given by the application. When the message handler stores a remote frame in the message object, it will store the received data length code. The data bytes of the message object will remain unchanged.

If the RxIE bit is set, the IntPnd bit will be set when a received remote frame is accepted and stored in the message object.

22.3.7.5 Configuration of a FIFO Buffer

With the exception of the EoB bit, the configuration of receive objects belonging to a FIFO buffer is the same as the configuration of a single receive object.

To concatenate multiple message objects to a FIFO buffer, the identifiers and masks (if used) of these message objects have to be programmed to matching values. Due to the implicit priority of the message objects, the message object with the lowest number will be the first message object of the FIFO buffer. The EoB bit of all message objects of a FIFO buffer except the last one have to be programmed to zero. The EoB bits of the last message object of a FIFO Buffer is set to one, configuring it as the end of the block.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.3.8 Message Handling

When initialization is finished, the DCAN module synchronizes itself to the traffic on the CAN bus. It does acceptance filtering on received messages and stores those frames that are accepted into the designated message objects. The application has to update the data of the messages to be transmitted and to enable and request their transmission. The transmission is requested automatically when a matching remote frame is received.

The application may read messages which are received and accepted. Messages that are not read before the next messages is accepted for the same message object will be overwritten.

Messages may be read interrupt-driven or after polling of NewDat.

22.3.8.1 Message Handler Overview

The message handler state machine controls the data transfer between the Rx/Tx shift register of the CAN core and the message RAM. It performs the following tasks:

- Data transfer from message RAM to CAN core (messages to be transmitted)
- Data transfer from CAN core to the message RAM (received messages)
- Data transfer from CAN core to the acceptance filtering unit
- Scanning of message RAM for a matching message object (acceptance filtering)
- Scanning the same message object after being changed by IF1/IF2 registers when priority is the same or higher as the message the object found by last scanning
- Handling of TxRqst flags
- Handling of interrupt flags

The message handler registers contains status flags of all message objects grouped into the following topics:

- Transmission Request Flags
- New Data Flags
- Interrupt Pending Flags
- Message Valid Registers

Instead of collecting above listed status information of each message object via IFx registers separately, these message handler registers provides a fast and easy way to get an overview, for example, about all pending transmission requests.

All message handler registers are read-only.

22.3.8.2 Receive/Transmit Priority

The receive/transmit priority for the message objects is attached to the message number, not to the CAN identifier. Message object 1 has the highest priority, while the last implemented message object has the lowest priority. If more than one transmission request is pending, they are serviced due to the priority of the corresponding message object so messages with the highest priority, for example, can be placed in the message objects with the lowest numbers.

The acceptance filtering for received data frames or remote frames is also done in ascending order of message objects, so a frame that has been accepted by a message object cannot be accepted by another message object with a higher message number. The last message object may be configured to accept any data frame or remote frame that was not accepted by any other message object, for nodes that need to log the complete message traffic on the CAN bus.

22.3.8.3 Transmission of Messages in Event Driven CAN Communication

If the shift register of the CAN core is ready for loading and if there is no data transfer between the IFx registers and message RAM, the d bits in the Message Valid register (DCAN MSGVAL12 to DCAN MSGVAL78) and the TxRqst bits in the transmission request register are evaluated. The valid message object with the highest priority pending transmission request is loaded into the shift register by the message handler and the transmission is started. The message object's NewDat bit is reset.

Copyright © 2011, Texas Instruments Incorporated

After a successful transmission and if no new data was written to the message object (NewDat = ‘0’) since the start of the transmission, the TxRqst bit will be reset. If TxIE is set, IntPnd will be set after a successful transmission. If the DCAN has lost the arbitration or if an error occurred during the transmission, the message will be retransmitted as soon as the CAN bus is free again. If meanwhile the transmission of a message with higher priority has been requested, the messages will be transmitted in the order of their priority.

If automatic retransmission mode is disabled by setting the DAR bit in the CAN control register, the behavior of bits TxRqst and NewDat in the Message Control register of the interface register set is as follows:

- When a transmission starts, the TxRqst bit of the respective interface register set is reset, while bit NewDat remains set.
- When the transmission has been successfully completed, the NewDat bit is reset.

When a transmission failed (lost arbitration or error) bit NewDat remains set. To restart the transmission, the application has to set TxRqst again.

Received remote frames do not require a receive object. They will automatically trigger the transmission of a data frame, if in the matching transmit object the RmtEn bit is set.

22.3.8.4 Updating a Transmit Object

The CPU may update the data bytes of a transmit object any time via the IF1/IF2 interface registers, neither d nor TxRqst have to be reset before the update.

Even if only a part of the data bytes are to be updated, all four bytes in the corresponding IF1/IF2 Data A register (DCAN IF1DATA/DCAN IF2DATA) or IF1/IF2 Data B register (DCAN IF1DATB/DCAN IF2DATB) have to be valid before the content of that register is transferred to the message object. Either the CPU has to write all four bytes into the IF1/IF2 data register or the message object is transferred to the IF1/IF2 data register before the CPU writes the new data bytes.

When only the data bytes are updated, first 0x87 can be written to bits [23:16] of the IF1/IF2 Command register (DCAN IF1CMD/DCAN IF2CMD) and then the number of the message object is written to bits [7:0] of the command register, concurrently updating the data bytes and setting TxRqst with NewDat.

To prevent the reset of TxRqst at the end of a transmission that may already be in progress while the data is updated, NewDat has to be set together with TxRqst in event driven CAN communication. For details, see Section 22.3.8.3.

When NewDat is set together with TxRqst, NewDat will be reset as soon as the new transmission has started.

22.3.8.5 Changing a Transmit Object

If the number of implemented message objects is not sufficient to be used as permanent message objects only, the transmit objects may be managed dynamically. The CPU can write the whole message (arbitration, control, and data) into the interface register. The bits [23:16] of the command register can be set to 0xB7 for the transfer of the whole message object content into the message object. Neither d nor TxRqst have to be reset before this operation.

If a previously requested transmission of this message object is not completed but already in progress, it will be continued; however, it will not be repeated if it is disturbed.

To only update the data bytes of a message to be transmitted, bits [23:16] of the command register should be set to 0x87.

NOTE: After the update of the transmit object, the interface register set will contain a copy of the actual contents of the object, including the part that had not been updated.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.3.8.6 Acceptance Filtering of Received Messages

When the arbitration and control bits (Identifier + IDE + RTR + DLC) of an incoming message are completely shifted into the shift register of the CAN core, the message handler starts the scan of the message RAM for a matching valid message object:

- The acceptance filtering unit is loaded with the arbitration bits from the CAN core shift register.
- Then the arbitration and mask bits (including MsgVal, UMask, NewDat, and EoB) of message object 1 are loaded into the acceptance filtering unit and are compared with the arbitration bits from the shift register. This is repeated for all following message objects until a matching message object is found, or until the end of the message RAM is reached.
- If a match occurs, the scanning is stopped and the message handler proceeds depending on the type of the frame (data frame or remote frame) received.

22.3.8.7 Reception of Data Frames

The message handler stores the message from the CAN core shift register into the respective message object in the message RAM. Not only the data bytes, but all arbitration bits and the data length code are stored into the corresponding message object. This ensures that the data bytes stay associated to the identifier even if arbitration mask registers are used.

The NewDat bit is set to indicate that new data (not yet seen by the CPU) has been received. The CPU should reset the NewDat bit when it reads the message object. If at the time of the reception the NewDat bit was already set, MsgLst is set to indicate that the previous data (supposedly not seen by the CPU) is lost. If the RxIE bit is set, the IntPnd bit is set, causing the interrupt register to point to this message object.

The TxRqst bit of this message object is reset to prevent the transmission of a remote frame, while the requested data frame has just been received.

22.3.8.8 Reception of Remote Frames

When a remote frame is received, three different configurations of the matching message object have to be considered:

- Dir = ‘1’ (direction = transmit), RmtEn = ‘1’, UMask = ‘1’ or ‘0’

  The TxRqst bit of this message object is set at the reception of a matching remote frame. The rest of the message object remains unchanged.
- Dir = ‘1’ (direction = transmit), RmtEn = ‘0’, UMask = ‘0’

  The remote frame is ignored, this message object remains unchanged.
- Dir = ‘1’ (direction = transmit), RmtEn = ‘0’, UMask = ‘1’

  The remote frame is treated similar to a received data frame. At the reception of a matching Remote Message Handling October 2009 Controller Area Network (DCAN) Module 371 www.ti.com Frame, the TxRqst bit of this message object is reset. The arbitration and control bits (Identifier + IDE + RTR + DLC) from the shift register are stored in the message object in the message RAM and the NewDat bit of this message object is set. The data bytes of the message object remain unchanged.

22.3.8.9 Reading Received Messages

The CPU may read a received message any time via the IFx interface register. The data consistency is guaranteed by the message handler state machine. Typically the CPU will write first 0x7F to bits [23:16] and then the number of the message object to bits [7:0] of the command register. That combination will transfer the entire received message from the message RAM into the interface register set. Additionally, the bits NewDat and IntPnd are cleared in the message RAM (not in the interface register set). The values of these bits in the message control register always reflect the status before resetting the bits. If the message object uses masks for acceptance filtering, the arbitration bits show which of the different matching messages has been received.

The actual value of NewDat shows whether a new message has been received since last time when this message object was read. The actual value of MsgLst shows whether more than one message has been received since the last time when this message object was read. MsgLst will not be automatically reset.

Copyright © 2011, Texas Instruments Incorporated

22.3.8.10 Requesting New Data for a Receive Object

By means of a remote frame, the CPU may request another CAN node to provide new data for a receive object. Setting the TxRqst bit of a receive object will cause the transmission of a remote frame with the identifier of the receive object. This remote frame triggers the other CAN node to start the transmission of the matching data frame. If the matching data frame is received before the remote frame could be transmitted, the TxRqst bit is automatically reset. Setting the TxRqst bit without changing the contents of a message object requires the value 0x84 in bits [23:16] of the command register.

22.3.8.11 Storing Received Messages in FIFO Buffers

Several message objects may be grouped to form one or more FIFO buffers. Each FIFO buffer configured to store received messages with a particular (group of) identifier(s). arbitration and mask registers of the FIFO buffer's message objects are identical. The end of buffer (EoB) bits of all but the last of the FIFO buffer's message objects are '0'; in the last one the EoB bit is '1.'

Received messages with identifiers matching to a FIFO buffer are stored into a message object of this FIFO buffer, starting with the message object with the lowest message number. When a message is stored into a message object of a FIFO buffer, the NewDat bit of this message object is set. By setting NewDat while EoB is '0', the message object is locked for further write accesses by the message handler until the CPU has cleared the NewDat bit.

Messages are stored into a FIFO buffer until the last message object of this FIFO buffer is reached. If none of the preceding message objects is released by writing NewDat to '0,' all further messages for this FIFO buffer will be written into the last message object of the FIFO buffer (EoB = '1') and therefore overwrite previous messages in this message object.

22.3.8.12 Reading From a FIFO Buffer

Several messages may be accumulated in a set of message objects which are concatenated to form a FIFO buffer before the application program is required (in order to avoid the loss of data) to empty the buffer.

A FIFO buffer of length N will store –1 plus the last received message since last time it was cleared.

A FIFO buffer is cleared by reading and resetting the NewDat bits of all its message objects, starting at the FIFO Object with the lowest message number. This should be done in a subroutine following the example shown in Figure 22-12.

NOTE: All message objects of a FIFO buffer needs to be read and cleared before the next batch of messages can be stored. Otherwise, true FIFO functionality can not be guaranteed, since the message objects of a partly read buffer will be re-filled according to the normal (descending) priority.

Reading from a FIFO buffer message object and resetting its NewDat bit is handled the same way as reading from a single message object.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 22-12. CPU Handling of a FIFO Buffer (Interrupt Driven)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.3.9 CAN Bit Timing

The DCAN supports bit rates between less than 1 kBit/s and 1000 kBit/s.

Each member of the CAN network has its own clock generator, typically derived from a crystal oscillator. The bit timing parameters can be configured individually for each CAN node, creating a common bit rate even though the CAN nodes' oscillator periods (f_{osc}) may be different.

The frequencies of these oscillators are not absolutely stable. Small variations are caused by changes in temperature or voltage and by deteriorating components. As long as the variations remain inside a specific oscillator tolerance range (df), the CAN nodes are able to compensate for the different bit rates by resynchronizing to the bit stream.

In many cases, the CAN bit synchronization will amend a faulty configuration of the CAN bit timing to such a degree that only occasionally an error frame is generated. In the case of arbitration however, when two or more CAN nodes simultaneously try to transmit a frame, a misplaced sample point may cause one of the transmitters to become error passive.

The analysis of such sporadic errors requires a detailed knowledge of the CAN bit synchronization inside a CAN node and of the CAN nodes' interaction on the CAN bus.

Even if minor errors in the configuration of the CAN bit timing do not result in immediate failure, the performance of a CAN network can be reduced significantly.

22.3.9.1 Bit Time and Bit Rate

According to the CAN specification, the bit time is divided into four segments (see Figure 22-13):

- Synchronization segment (Sync_Seg)
- Propagation time segment (Prop_Seg)
- Phase buffer segment 1 (Phase_Seg1)
- Phase buffer segment 2 (Phase_Seg2)

Figure 22-13. Bit Timing

Each segment consists of a specific number of time quanta. The length of one time quantum (t_q), which is the basic time unit of the bit time, is given by the CAN_CLK and the baud rate prescalers (BRPE and BRP). With these two baud rate prescalers combined, divider values from 1 to 1024 can be programmed:

t_q = Baud Rate Prescaler / CAN_CLK

Apart from the fixed length of the synchronization segment, these numbers are programmable. Table 22-7 describes the minimum programmable ranges required by the CAN protocol.

A given bit rate may be met by different bit time configurations.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 22-7. Parameters of the CAN Bit Time

Parameter	Range	Remark
Sync_Seg	1 t_q (fixed)	Synchronization of bus input to CAN_CLK
Prop_Seg	[1 … 8] t_q	Compensates for the physical delay times
Phase_Seg1	[1 … 8] t_q	May be lengthened temporarily by synchronization
Phase_Seg2	[1 … 8] t_q	May be shortened temporarily by synchronization
Synchronization Jump Width (SJW)	[1 … 4] t_q	May not be longer than either Phase Buffer Segment

NOTE: For proper functionality of the CAN network, the physical delay times and the oscillator's tolerance range have to be considered.

22.3.9.1.1 Synchronization Segment

The synchronization segment (Sync_Seg) is the part of the bit time where edges of the CAN bus level are expected to occur. If an edge occurs outside of Sync_Seg, its distance to the Sync_Seg is called the phase error of this edge.

22.3.9.1.2 Propagation Time Segment

This part of the bit time is used to compensate physical delay times within the CAN network. These delay times consist of the signal propagation time on the bus and the internal delay time of the CAN nodes.

Any CAN node synchronized to the bit stream on the CAN bus can be out of phase with the transmitter of the bit stream, caused by the signal propagation time between the two nodes. The CAN protocol's nondestructive bitwise arbitration and the dominant acknowledge bit provided by receivers of CAN messages require that a CAN node transmitting a bit stream must also be able to receive dominant bits transmitted by other CAN nodes that are synchronized to that bit stream. The example in Figure 22-14 shows the phase shift and propagation times between two CAN nodes.

Figure 22-14. The Propagation Time Segment

Delay A_to_B >= node output delay(A) + bus line delay(A→B) + node input delay(B)

Prop_Seg >= Delay A_to_B + Delay B_to_A

Prop_Seg >= 2 • [max(node output delay+bus line delay + node input delay)]

In this example, both nodes A and B are transmitters performing an arbitration for the CAN bus. The node A has sent its start of frame bit less than one bit time earlier than node B, therefore node B has

Copyright © 2011, Texas Instruments Incorporated

synchronized itself to the received edge from recessive to dominant. Since node B has received this edge delay(A_to_B) after it has been transmitted, node B's bit timing segments are shifted with regard to node A. Node B sends an identifier with higher priority and so it will win the arbitration at a specific identifier bit when it transmits a dominant bit while node A transmits a recessive bit. The dominant bit transmitted by node B will arrive at node A after the delay (B_to_A).

Due to oscillator tolerances, the actual position of node A's sample point can be anywhere inside the nominal range of node A's Phase Buffer Segments, so the bit transmitted by node B must arrive at node A before the start of Phase_Seg1. This condition defines the length of Prop_Seg.

If the edge from recessive to dominant transmitted by node B would arrive at node A after the start of Phase_Seg1, it could happen that node A samples a recessive bit instead of a dominant bit, resulting in a bit error and the destruction of the current frame by an error flag.

This error only occurs when two nodes arbitrate for the CAN bus which have oscillators of opposite ends of the tolerance range and are separated by a long bus line; this is an example of a minor error in the bit timing configuration (Prop_Seg too short) that causes sporadic bus errors.

Some CAN implementations provide an optional 3 Sample Mode. The DCAN does not. In this mode, the CAN bus input signal passes a digital low-pass filter, using three samples and a majority logic to determine the valid bit value. This results in an additional input delay of 1 t_q, requiring a longer Prop_Seg.

22.3.9.1.3 Phase Buffer Segments and Synchronization

The phase buffer segments (Phase_Seg1 and Phase_Seg2) and the synchronization jump width (SJW) are used to compensate for the oscillator tolerance.

The phase buffer segments surround the sample point and may be lengthened or shortened by synchronization.

The synchronization jump width (SJW) defines how far the resynchronizing mechanism may move the sample point inside the limits defined by the phase buffer segments to compensate for edge phase errors.

Synchronizations occur on edges from recessive to dominant. Their purpose is to control the distance between edges and sample points.

Edges are detected by sampling the actual bus level in each time quantum and comparing it with the bus level at the previous sample point. A synchronization may be done only if a recessive bit was sampled at the previous sample point and if the actual time quantum's bus level is dominant.

An edge is synchronous if it occurs inside of Sync_Seg; otherwise, its distance to the Sync_Seg is the edge phase error, measured in time quanta. If the edge occurs before Sync_Seg, the phase error is negative, else it is positive.

Two types of synchronization exist: hard synchronization and resynchronizing. A hard synchronization is done once at the start of a frame; inside a frame, only resynchronization is possible.

- Hard Synchronization

  After a hard synchronization, the bit time is restarted with the end of Sync_Seg, regardless of the edge phase error. Thus hard synchronization forces the edge which has caused the hard synchronization, to lie within the synchronization segment of the restarted bit time.

- Bit Resynchronizations

  Resynchronization leads to a shortening or lengthening of the Bit time such that the position of the sample point is shifted with regard to the edge.

  When the phase error of the edge which causes resynchronization is positive, Phase_Seg1 is lengthened. If the magnitude of the phase error is less than SJW, Phase_Seg1 is lengthened by the magnitude of the phase error, else it is lengthened by SJW.

  When the phase error of the edge which causes Resynchronization is negative, Phase_Seg2 is shortened. If the magnitude of the phase error is less than SJW, Phase_Seg2 is shortened by the magnitude of the phase error, else it is shortened by SJW.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

If the magnitude of the phase error of the edge is less than or equal to the programmed value of SJW, the results of hard synchronization and resynchronization are the same. If the magnitude of the phase error is larger than SJW, the resynchronization cannot compensate the phase error completely, and an error of (phase error - SJW) remains.

Only one synchronization may be done between two sample points. The synchronizations maintain a minimum distance between edges and sample points, giving the bus level time to stabilize and filtering out spikes that are shorter than (Prop_Seg + Phase_Seg1).

Apart from noise spikes, most synchronizations are caused by arbitration. All nodes synchronize “hard” on the edge transmitted by the “leading” transceiver that started transmitting first, but due to propagation delay times, they cannot become ideally synchronized. The leading transmitter does not necessarily win the arbitration; therefore, the receivers have to synchronize themselves to different transmitters that subsequently take the lead and that are differently synchronized to the previously leading transmitter. The same happens at the acknowledge field, where the transmitter and some of the receivers will have to synchronize to that receiver that takes the lead in the transmission of the dominant acknowledge bit.

Synchronizations after the end of the arbitration will be caused by oscillator tolerance, when the differences in the oscillator’s clock periods of transmitter and receivers sum up during the time between synchronizations (at most ten bits). These summarized differences may not be longer than the SJW, limiting the oscillator’s tolerance range.

Figure 22-15 shows how the phase buffer segments are used to compensate for phase errors. There are three drawings of each two consecutive bit timings. The upper drawing shows the synchronization on a “late” edge, the lower drawing shows the synchronization on an “early” edge, and the middle drawing is the reference without synchronization.

Figure 22-15. Synchronization on Late and Early Edges

In the first example, an edge from recessive to dominant occurs at the end of Prop_Seg. The edge is "late" since it occurs after the Sync_Seg. Reacting to the late edge, Phase_Seg1 is lengthened so that the distance from the edge to the sample point is the same as it would have been from the Sync_Seg to the sample point if no edge had occurred. The phase error of this late edge is less than SJW, so it is fully compensated and the edge from dominant to recessive at the end of the bit, which is one nominal bit time long, occurs in the Sync_Seg.

In the second example, an edge from recessive to dominant occurs during Phase_Seg2. The edge is "early" since it occurs before a Sync_Seg. Reacting to the early edge, Phase_Seg2 is shortened and Sync_Seg is omitted, so that the distance from the edge to the sample point is the same as it would have been from a Sync_Seg to the sample point if no edge had occurred. As in the previous example, the magnitude of this early edge’s phase error is less than SJW, so it is fully compensated.

Copyright © 2011, Texas Instruments Incorporated

The phase buffer segments are lengthened or shortened temporarily only; at the next bit time, the segments return to their nominal programmed values.

In these examples, the bit timing is seen from the point of view of the CAN implementation's state machine, where the bit time starts and ends at the sample points. The state machine omits Sync_Seg when synchronizing on an early edge because it cannot subsequently redefine that time quantum of Phase_Seg2 where the edge occurs to be the Sync_Seg.

Figure 22-16 shows how short dominant noise spikes are filtered by synchronizations. In both examples, the spike starts at the end of Prop_Seg and has the length of (Prop_Seg + Phase_Seg1).

In the first example, the synchronization jump width is greater than or equal to the phase error of the spike's edge from recessive to dominant. Therefore the sample point is shifted after the end of the spike; a recessive bus level is sampled.

In the second example, SJW is shorter than the phase error, so the sample point cannot be shifted far enough; the dominant spike is sampled as actual bus level.

Figure 22-16. Filtering of Short Dominant Spikes

22.3.9.1.4 Oscillator Tolerance Range

With the introduction of CAN protocol version 1.2, the option to synchronize on edges from dominant to recessive became obsolete. Only edges from recessive to dominant are considered for synchronization. The protocol update to version 2.0 (A and B) had no influence on the oscillator tolerance.

The tolerance range df for an oscillator's frequency f_{osc} around the nominal frequency f_{nom} with:

$$(1-df)\bullet f_{nom} \le f_{osc} \le (1+df)\bullet f_{nom}$$

depends on the proportions of Phase_Seg1, Phase_Seg2, SJW, and the bit time. The maximum tolerance df is the defined by two conditions (both shall be met):

$$\text{I:} \quad df \le \frac{\min(TSeg1, TSeg2)}{2x(13x(bit_time - TSeg2))}$$

$$\text{II:} \quad df \le \frac{SJW}{20xbit_time}$$

It has to be considered that SJW may not be larger than the smaller of the phase buffer segments and that the propagation time segment limits that part of the bit time that may be used for the phase buffer segments.

Copyright © 2011, Texas Instruments Incorporated

The combination Prop_Seg = 1 and Phase_Seg1 = Phase_Seg2 = SJW = 4 allows the largest possible oscillator tolerance of 1.58%. This combination with a propagation time segment of only 10% of the bit time is not suitable for short bit times; it can be used for bit rates of up to 125 kBit/s (bit time = 8 µs) with a bus length of 40 m.

22.3.9.2 DCAN Bit Timing Registers

In the DCAN, the bit timing configuration is programmed in two register bytes, additionally a third byte for a baud rate prescaler extension of four bits (BREP) is provided. The sum of Prop_Seg and Phase_Seg1 (as TSEG1) is combined with Phase_Seg2 (as TSEG2) in one byte, SJW and BRP (plus BRPE in third byte) are combined in the other byte (see Figure 22-17).

Figure 22-17. Structure of the CAN Core's CAN Protocol Controller

In this bit timing register, the components TSEG1, TSEG2, SJW and BRP have to be programmed to a numerical value that is one less than its functional value; so instead of values in the range of [1…n], values in the range of [0…–1] are programmed. That way, e.g., SJW (functional range of [1…4]) is represented by only two bits.

Therefore, the length of the bit time is (programmed values) [TSEG1 + TSEG2 + 3] t_q or (functional values) [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] t_q.

The data in the bit timing register (BTR) is the configuration input of the CAN protocol controller. The baud rate prescaler (configured by BRPE/BRP) defines the length of the time quantum (the basic time unit of the bit time); the bit timing logic (configured by TSEG1, TSEG2, and SJW) defines the number of time quanta in the bit time.

The processing of the bit time, the calculation of the position of the sample point, and occasional synchronizations are controlled by the bit timing state machine, which is evaluated once each time quantum. The rest of the CAN protocol controller, the bit stream processor (BSP) state machine, is evaluated once each bit time, at the sample point.

The shift register serializes the messages to be sent and parallelizes received messages. Its loading and shifting is controlled by the BSP. The BSP translates messages into frames and vice versa. It generates and discards the enclosing fixed format bits, inserts and extracts stuff bits, calculates and checks the CRC code, performs the error management, and decides which type of synchronization is to be used. It is evaluated at the sample point and processes the sampled bus input bit. The time after the sample point that is needed to calculate the next bit to be sent (e.g., data bit, CRC bit, stuff bit, error flag, or idle) is called the information processing time (IPT), which is 0 tq for the DCAN.

Copyright © 2011, Texas Instruments Incorporated

Generally, the IPT is CAN controller-specific, but may not be longer than 2 tq. The IPC length is the lower limit of the programmed length of Phase_Seg2. In case of a synchronization, Phase_Seg2 may be shortened to a value less than IPT, which does not affect bus timing.

22.3.9.2.1 Calculation of the Bit Timing Parameters

Usually, the calculation of the bit timing configuration starts with a desired bit rate or bit time. The resulting bit time (1 / Bit rate) must be an integer multiple of the CAN clock period.

NOTE: 8 MHz is the minimum CAN clock frequency required to operate the DCAN at a bit rate of 1 MBit/s.

The bit time may consist of 8 to 25 time quanta. The length of the time quantum t_q is defined by the baud rate prescaler with t_q = (Baud Rate Prescaler) / CAN_CLK. Several combinations may lead to the desired bit time, allowing iterations of the following steps.

First part of the bit time to be defined is the Prop_Seg. Its length depends on the delay times measured in the system. A maximum bus length as well as a maximum node delay has to be defined for expandible CAN bus systems. The resulting time for Prop_Seg is converted into time quanta (rounded up to the nearest integer multiple of t_q).

The Sync_Seg is 1 t_q long (fixed), leaving (bit time – Prop_Seg – 1) t_q for the two Phase Buffer Segments. If the number of remaining t_q is even, the Phase Buffer Segments have the same length, Phase_Seg2 = Phase_Seg1, else Phase_Seg2 = Phase_Seg1 + 1.

The minimum nominal length of Phase_Seg2 has to be regarded as well. Phase_Seg2 may not be shorter than any CAN controller's Information Processing Time in the network, which is device dependent and can be in the range of [0…2] t_q.

The length of the synchronization jump width is set to its maximum value, which is the minimum of four (4) and Phase_Seg1.

The oscillator tolerance range necessary for the resulting configuration is calculated by the formulas given in Section 22.3.9.2.3.

If more than one configurations are possible to reach a certain Bit rate, it is recommended to choose the configuration which allows the highest oscillator tolerance range.

CAN nodes with different clocks require different configurations to come to the same bit rate. The calculation of the propagation time in the CAN network, based on the nodes with the longest delay times, is done once for the whole network.

The CAN system's oscillator tolerance range is limited by the node with the lowest tolerance range.

The calculation may show that bus length or bit rate have to be decreased or that the oscillator frequencies' stability has to be increased in order to find a protocol compliant configuration of the CAN bit timing.

The resulting configuration is written into the bit timing register:

Tseg2 = Phase_Seg2-1

Tseg1 = Phase_Seg1+ Prop_Seg-1

SJW = SynchronizationJumpWidth-1

BRP = Prescaler-1

22.3.9.2.2 Example for Bit Timing at High Baud Rate

In this example, the frequency of CAN_CLK is 10 MHz, BRP is 0, the bit rate is 1 MBit/s.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

t_q	100 ns	=	t_{CAN_CLK}
delay of bus driver	60 ns	=	
delay of receiver circuit	40 ns	=	
delay of bus line (40 m)	220 ns	=	
t_{Prop}	700 ns	=	INT (2*delays + 1) = 7 • t_q
t_{SJW}	100 ns	=	1 • t_q
t_{TSeg1}	800 ns	=	t_{Prop} + t_{SJW}
t_{TSeg2}	100 ns	=	Information Processing Time + 1 • t_q
$t_{Sync\text{-}Seg}$	100 ns	=	1 • t_q
bit time	1000 ns	=	$t_{Sync\text{-}Seg}$ + t_{TSeg1} + t_{TSeg2}
tolerance for CAN_CLK	0.43 %	=	$\frac{(\min(TSeg1, TSeg2))}{2x(13x(bit_time - TSeg2))}$
		=	$\frac{0.1\,\mu s}{2x(13x(1\,\mu s - 0.1\,\mu s))}$

In this example, the concatenated bit time parameters are $(1\text{-}1)_3$&$(8\text{-}1)_4$&$(1\text{-}1)_2$&$(1\text{-}1)_6$, so the bit timing register is programmed to = 00000700.

22.3.9.2.3 Example for Bit Timing at Low Baud Rate

In this example, the frequency of CAN_CLK is 2 MHz, BRP is 1, the bit rate is 100 KBit/s.

t_q	1 µs	=	t_{CAN_CLK}
Delay of bus driver	200 ns	=	
Delay of receiver circuit	80 ns	=	
Delay of bus line (40 m)	220 ns	=	
t_{Prop}	1 µs	=	1 • t_q
t_{SJW}	4 µs	=	4 • t_q
t_{TSeg1}	5 µs	=	t_{Prop} + t_{SJW}
t_{TSeg2}	3 µs	=	Information Processing Time + 3 • t_q
$t_{Sync\text{-}Seg}$	1 µs	=	1 • t_q
Bit time	9 µs	=	$t_{Sync\text{-}Seg}$ + t_{TSeg1} + t_{TSeg2}
Tolerance for CAN_CLK		=	$\frac{(\min(TSeg1, TSeg2))}{2x(13x(bit_time - TSeg2))}$
		=	$\frac{0.1\,\mu s}{2x(13x(1\,\mu s - 0.1\,\mu s))}$

In this example, the concatenated bit time parameters are $(3\text{-}1)_3$&$(5\text{-}1)_4$&$(4\text{-}1)_2$&$(2\text{-}1)_6$, so the bit timing register is programmed to = 0x000024C1.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.3.10 Message Interface Register Sets

The interface register sets control the CPU read and write accesses to the message RAM. There are two interface registers sets for read/write access, IF1 and IF2 and one interface register set for read access only, IF3.

Due to the structure of the message RAM, it is not possible to change single bits or bytes of a message object. Instead, always a complete message object in the message RAM is accessed. Therefore the data transfer from the IF1/IF2 registers to the message RAM requires the message handler to perform a read-modify-write cycle: First those parts of the message object that are not to be changed are read from the message RAM into the interface register set, and after the update the whole content of the interface register set is written into the message object.

After the partial write of a message object, those parts of the interface register set that are not selected in the command register, will be set to the actual contents of the selected message object.

After the partial read of a message object, those parts of the interface register set that are not selected in the command register, will be left unchanged.

By buffering the data to be transferred, the interface register sets avoid conflicts between concurrent CPU accesses to the message RAM and CAN message reception and transmission. A complete message object (see Section 22.3.11.1) or parts of the message object may be transferred between the message RAM and the IF1/IF2 register set (see Section 22.5.19) in one single transfer. This transfer, performed in parallel on all selected parts of the message object, guarantees the data consistency of the CAN message.

22.3.10.1 Message Interface Register Sets 1 and 2

The IF1 and IF2 register sets control the data transfer to and from the message object. The command register addresses the desired message object in the message RAM and specifies whether a complete message object or only parts should be transferred. The data transfer is initiated by writing the message number to the bits [7:0] of the command register.

When the CPU initiates a data transfer between the IF1/IF2 registers and message RAM, the message handler sets the busy bit in the respective command register to ‘1’. After the transfer has completed, the busy bit is set back to ‘0’ (see Figure 22-18).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 22-18. Data Transfer Between IF1/IF2 Registers and Message RAM

22.3.10.2 IF3 Register Set

The IF3 register set can automatically be updated with received message objects without the need to initiate the transfer from message RAM by CPU. The intention of this feature of IF3 is to provide an interface for the DMA to read packets efficiently. The automatic update functionality can be programmed for each message object (Section 22.5.28).

All valid message objects in message RAM which are configured for automatic update, will be checked for active NewDat flags. If such a message object is found, it will be transferred to the IF3 register (if no previous DMA transfers are ongoing), controlled by IF3 Observation register. If more than one NewDat flag is active, the message object with the lowest number has the highest priority for automatic IF3 update.

The NewDat bit in the message object will be reset by a transfer to IF3.

If DCAN internal IF3 update is complete, a DMA request is generated. The DMA request stays active until first read access to one of the IF3 registers. The DMA functionality has to be enabled by setting bit DE3 in CAN control register. Please refer to the device datasheet to find out if this DMA source is available.

NOTE: The IF3 register set can not be used for transferring data into message objects.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.3.11 Message RAM

The DCAN message RAM contains message objects and parity bits for the message objects. There are up to 64 message objects in the message RAM.

During normal operation, accesses to the message RAM are performed via the interface register sets, and the CPU cannot directly access the message RAM.

The interface register sets IF1 and IF2 provide indirect read/write access from the CPU to the message RAM. The IF1 and IF2 register sets can buffer control and user data to be transferred to and from the message objects.

The third interface register set IF3 can be configured to automatically receive control and user data from the message RAM when a message object has been updated after reception of a CAN message. The CPU does not need to initiate the transfer from message RAM to IF3 register set.

The message handler avoids potential conflicts between concurrent accesses to message RAM and CAN frame reception/transmission.

There are two modes where the message RAM can be directly accessed by the CPU:

- Debug/Suspend mode (see Section 22.3.11.3)
- RAM Direct Access (RDA) mode (see Section 22.3.11.4)

For the message RAM base address, see the device-specific data sheet.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.3.11.1 Structure of Message Objects

Table 22-8 shows the structure of a message object.

The grayed fields are those parts of the message object which are represented in dedicated registers. For example, the transmit request flags of all message objects are represented in centralized transmit request registers.

Table 22-8. Structure of a Message Object

Message Object												
UMask	**Msk[28:0]**	**MXtd**	**MDir**	**EoB**	**unused**	**NewDat**	**MsgLst**	**RxIE**	**TxIE**	**IntPnd**	**RmtEn**	**TxRqst**
MsgVal	ID[28:0]	Xtd	Dir	DLC[3:0]	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7

Table 22-9. Field Descriptions

Name	Value	Description
MsgVal		Message valid
	0	The message object is ignored by the message handler.
	1	The message object is to be used by the message handler.
		Note: This bit may be kept at level ‘1’ even when the identifier bits ID[28:0], the control bits Xtd, Dir, or the data length code are changed. It should be reset if the Messages Object is no longer required.
UMask		Use acceptance mask
	0	Mask bits (Msk[28:0], MXtd and MDir) are ignored and not used for acceptance filtering.
	1	Mask bits are used for acceptance filtering.
		Note: If the UMask bit is set to one, the message object’s mask bits have to be programmed during initialization of the message object before MsgVal is set to one.
ID[28:0]		Message identifier
	ID[28:0]	29-bit (“extended”) identifier bits
	ID[28:18]	11-bit (“standard”) identifier bits
Msk[28:0]		Identifier mask
	0	The corresponding bit in the message identifier is not used for acceptance filtering (don’t care).
	1	The corresponding bit in the message identifier is used for acceptance filtering.
Xtd		Mask extended identifier
	0	The extended identifier bit (IDE) has no effect on the acceptance filtering.
	1	The extended identifier bit (IDE) is used for acceptance filtering.
		Note: When 11-bit (“standard”) Identifiers are used for a message object, the identifiers of received data frames are written into bits ID[28:18]. For acceptance filtering, only these bits together with mask bits Msk[28:18] are considered.
Dir		Message direction
	0	Direction = receive: On TxRqst, a remote frame with the identifier of this message object is transmitted. On reception of a data frame with matching identifier, the message is stored in this message object.
	1	Direction = transmit: On TxRqst, a data frame is transmitted. On reception of a remote frame with matching identifier, the TxRqst bit of this message object is set (if RmtEn = one).
MDir		Mask message direction
	0	The message direction bit (Dir) has no effect on the acceptance filtering.
	1	The message direction bit (Dir) is used for acceptance filtering.
EOB		End of block
	0	The message object is part of a FIFO Buffer block and is not the last message object of this FIFO Buffer block.
	1	The message object is a single message object or the last message object in a FIFO Buffer Block.
		Note: This bit is used to concatenate multiple message objects to build a FIFO Buffer. For single message objects (not belonging to a FIFO Buffer), this bit must always be set to one.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 22-9. Field Descriptions (continued)

Name	Value	Description
NewDat		New data
	0	No new data has been written into the data bytes of this message object by the message handler since the last time when this flag was cleared by the CPU.
	1	The message handler or the CPU has written new data into the data bytes of this message object.
MsgLst		Message lost (only valid for message objects with direction = receive)
	0	No message was lost since the last time when this bit was reset by the CPU.
	1	The message handler stored a new message into this message object when NewDat was still set, so the previous message has been overwritten.
RxIE		Receive interrupt enable
	0	IntPnd will not be triggered after the successful reception of a frame.
	1	IntPnd will be triggered after the successful reception of a frame.
TxIE		Transmit interrupt enable
	0	IntPnd will not be triggered after the successful transmission of a frame.
	1	IntPnd will be triggered after the successful transmission of a frame.
IntPnd		Interrupt pending
	0	This message object is not the source of an interrupt.
	1	This message object is the source of an interrupt. The interrupt Identifier in the interrupt register will point to this message object if there is no other interrupt source with higher priority.
RmtEn		Remote enable
	0	At the reception of a remote frame, TxRqst is not changed.
	1	At the reception of a remote frame, TxRqst is set.
TxRqst		Transmit request
	0	This message object is not waiting for a transmission.
	1	The transmission of this message object is requested and is not yet done.
DLC[3:0]		Data length code
	0	Data frame has 0-8 data bits.
	1	Data frame has 8 data bytes.
		Note: The data length code of a message object must be defined to the same value as in the corresponding objects with the same identifier at other nodes. When the message handler stores a data frame, it will write the DLC to the value given by the received message.
Data 0		1st data byte of a CAN data frame
Data 1		2nd data byte of a CAN data frame
Data 2		3rd data byte of a CAN data frame
Data 3		4th data byte of a CAN data frame
Data 4		5th data byte of a CAN data frame
Data 5		6th data byte of a CAN data frame
Data 6		7th data byte of a CAN data frame
Data 7		8th data byte of a CAN data frame
		Note: Byte Data 0 is the first data byte shifted into the shift register of the CAN core during a reception, byte Data 7 is the last. When the message handler stores a data frame, it will write all the eight data bytes into a message object. If the data length code is less than 8, the remaining bytes of the message object may be overwritten by undefined values.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.3.11.2 Addressing Message Objects in RAM

The starting location of a particular message object in RAM is:
Message RAM base address + (message object number) * 0x20

This means that message object 1 starts at offset 0x0020; message object 2 starts at offset 0x0040, etc.

'0' is not a valid message object number.

The base address for DCAN0 RAM is 0x481C_D000 and DCAN1 RAM is 0x481D_1000.

Message object number 1 has the highest priority.

Table 22-10. Message RAM addressing in Debug/Suspend and RDA Mode

Message Object Number	Offset From Base Address	Word Number	Debug/Suspend Mode, see Section 22.3.11.3	RDA mode, see Section 22.3.11.4
1	0x0020	1	Parity	Data Bytes 4-7
	0x0024	2	MXtd,MDir,Mask	Data Bytes 0-3
	0x0028	3	Xtd,Dir,ID	ID[27:0],DLC
	0x002C	4	Ctrl	Mask,Xtd,Dir,ID[28]
	0x0030	5	Data Bytes 3-0	Parity,Ctrl,MXtd,MDir
	0x0034	6	Data Bytes 7-4	…
…	…	…	…	…
31	0x03E0	1	Parity	Data Bytes 4-7
	0x03E4	2	MXtd,MDir,Mask	Data Bytes 0-3
	0x03E8	3	Xtd,Dir,ID	ID[27:0],DLC
	0x03EC	4	Ctrl	Mask,Xtd,Dir,ID[28]
	0x03F0	5	Data Bytes 3-0	Parity,Ctrl,MXtd,MDir
	0x03F4	6	Data Bytes 7-4	…
…	…	…	…	…
63	0x07E0	1	Parity	Data Bytes 4-7
	0x07E4	2	MXtd,MDir,Mask	Data Bytes 0-3
	0x07E8	3	Xtd,Dir,ID	ID[27:0],DLC
	0x07EC	4	Ctrl	Mask,Xtd,Dir,ID[28]
	0x07F0	5	Data Bytes 3-0	Parity,Ctrl,MXtd,MDir
	0x07F4	6	Data Bytes 7-4	…
last implemented (32(DCAN3) or 64)	0x0000	1	Parity	Data Bytes 4-7
	0x0004	2	MXtd,MDir,Mask	Data Bytes 0-3
	0x0008	3	Xtd,Dir,ID	ID[27:0],DLC
	0x000C	4	Ctrl	Mask,Xtd,Dir,ID[28]
	0x0010	5	Data Bytes 3-0	Parity,Ctrl,MXtd,MDir
	0x0014	6	Data Bytes 7-4	…

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.3.11.3 Message RAM Representation in Debug/Suspend Mode

In debug/suspend mode, the message RAM will be memory mapped. This allows the external debug unit to access the message RAM.

NOTE: During debug/suspend mode, the message RAM cannot be accessed via the IFx register sets.

Table 22-11. Message RAM Representation in Debug/Suspend Mode

Bit #	31/15	30/14	29/13	29/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
MsgAddr + 0x00	Reserved															
	Reserved												Parity[4:0]			
MsgAddr + 0x04	MXtd	MDir	Rsvd	Msk[28:16]												
	Msk[15:0]															
MsgAddr + 0x08	Rsvd	Xtd	Dir	ID[28:16]												
	ID[15:0]															
MsgAddr + 0x0C	Reserved															
	Rsvd	MsgLst	Rsvd	UMask	TxIE	RxTE	RmtEn	Rsvd	EOB	Reserved			DLC[3:0]			
MsgAddr + 0x10	Data 3								Data 2							
	Data 1								Data 0							
MsgAddr + 0x14	Data 7								Data 6							
	Data 5								Data 4							

22.3.11.4 Message RAM Representation in Direct Access Mode

When the RDA bit in the test register is set while the DCAN module is in test mode (test bit in the CAN control register is set), the CPU has direct access to the message RAM. Due to the 32-bit bus structure, the RAM is split into word lines to support this feature. The CPU has access to one word line at a time only.

In RAM direct access mode, the RAM is represented by a continuous memory space within the address frame of the DCAN module, starting at the message RAM base address.

Note: During direct access mode, the message RAM cannot be accessed via the IFx register sets.

Any read or write to the RAM addresses for RamDirectAccess during normal operation mode (TestMode bit or RDA bit not set) will be ignored.

Table 22-12. Message RAM Representation in RAM Direct Access Mode

Bit #	31/15	30/14	29/13	29/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
MsgAddr + 0x00	Data 4								Data 5							
	Data 6								Data 7							
MsgAddr + 0x04	Data 0								Data 1							
	Data 2								Data 3							
MsgAddr + 0x08	ID[27:12]															
	ID[11:0]												DLC[3:0]			
	Msk[28:13]															

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 22-12. Message RAM Representation in RAM Direct Access Mode (continued)

Bit #	31/ 15	30/ 14	29/ 13	29/ 12	27/ 11	26/ 10	25/ 9	24/ 8	23/ 7	22/ 6	21/ 5	20/ 4	19/ 3	18/ 2	17/ 1	16/ 0
MsgAddr + 0x0C	Msk[12:0]													Xtd	Dir	ID[28]
MsgAddr + 0x10	Reserved															Parity [4]
	MsgLst				Unused				Msg Lst	UMa sk	TxIE	RxT E	Rmt En	EOB	MX td	MDir

NOTE: Writes to unused bits has no effect.

22.3.12 GIO Support

The CAN_RX and CAN_TX pins of the DCAN module can be used as general purpose IO pins, if CAN functionality is not needed. This function is controlled by the CAN TX IO control register (DCAN TIOC) (see Section 22.5.29) and the CAN RX IO control register (DCAN RIOC) (see Section 22.5.30).

22.4 Use Cases

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.5 DCAN Control Registers

The base address for the:

- DCAN0 registers are 0x481C_C000.
- DCAN1 registers are 0x481D_0000.

After hardware reset, the registers of the DCAN hold the values shown in the register descriptions.

Additionally, the Bus-Off state is reset and the CAN_TX pin is set to recessive (HIGH). The Init bit in the CAN control register is set to enable the software initialization. The DCAN will not influence the CAN bus until the CPU resets Init to '0'.

Table 22-13. DCAN Control Register Summary Table

Offset	Acronym	Description	Section
0x00	DCAN CTL	CAN Control Register	Section 22.5.1
0x04	DCAN ES and PARITYERR_EOI	Error and Status Register Parity Error EOI Register	Section 22.5.2
0x08	DCAN ERRC	Error Counter Register	Section 22.5.3
0x0C	DCAN BTR	Bit Timing Register	Section 22.5.4
0x10	DCAN INT	Interrupt Register	Section 22.5.5
0x14	DCAN TEST	Test Register	Section 22.5.6
0x1C	DCAN PERR	Parity Error Code Register	Section 22.5.7
0x80	DCAN ABOTR	Auto-Bus-On Time Register	Section 22.5.8
0x84	DCAN TXRQ X	Transmission Request X Register	Section 22.5.9
0x88 to 0x94	DCAN TXRQ12 to DCAN TXRQ78	Transmission Request Registers	Section 22.5.10
0x98	DCAN NWDAT X	New Data X Register	Section 22.5.11
0x9C to 0xA8	DCAN NWDAT12 to DCAN NWDAT78	New Data Registers	Section 22.5.12
0xAC	DCAN INTPND X	Interrupt Pending X Register	Section 22.5.13
0xB0 to 0xBC	DCAN INTPND12 to DCAN INTPND78)	Interrupt Pending Registers	Section 22.5.14
0xC0	DCAN MSGVAL X	Message Valid X Register	Section 22.5.15
0xC4 to 0xD0	DCAN MSGVAL12 to DCAN MSGVAL78	Message Valid Registers	Section 22.5.16
0xD8 to 0xE4	DCAN INTMUX12 to DCAN INTMUX78	Interrupt Multiplexer Registers	Section 22.5.17
0x100	DCAN IF1CMD	IF1 Command Registers	Section 22.5.18
0x120	DCAN IF2CMD	IF2 Command Registers	Section 22.5.18
0x104	DCAN IF1MSK	IF1 Mask Register	Section 22.5.19
0x124	DCAN IF2MSK	IF2 Mask Register	Section 22.5.19
0x108	DCAN IF1ARB	IF1 Arbitration Register	Section 22.5.20
0x128	DCAN IF2ARB	IF2 Arbitration Register	Section 22.5.20
0x10C	DCAN IF1MCTL	IF1 Message Control Register	Section 22.5.21
0x12C	DCAN IF2MCTL	IF2 Message Control Register	Section 22.5.21
0x110	DCAN IF1DATA	IF1 Data A Register	Section 22.5.22
0x114	DCAN IF1DATB	IF1 Data B Register	Section 22.5.22
0x130	DCAN IF2DATA	IF2 Data A Register	Section 22.5.22
0x134	DCAN IF2DATB	IF2 Data B Register	Section 22.5.22
0x140	DCAN IF3OBS	IF3 Observation Register	Section 22.5.23
0x144	DCAN IF3MSK	IF3 Mask Register	Section 22.5.24
0x148	DCAN IF3ARB	IF3 Arbitration Register	Section 22.5.25
0x14C	DCAN IF3MCTL	IF3 Message Control Register	Section 22.5.26
0x150	DCAN IF3DATA	IF3 Data A Register	Section 22.5.27

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 22-13. DCAN Control Register Summary Table (continued)

Offset	Acronym	Description	Section
0x154	DCAN IF3DATB	IF3 Data B Register	Section 22.5.27
0x160 to 0x16C	DCAN IF3UPD12 to IF3UPD78	IF3 Update Enable Registers	Section 22.5.28
0x1E0	DCAN TIOC	CAN TX IO Control Register	Section 22.5.29
0x1E4	DCAN RIOC	CAN RX IO Control Register	Section 22.5.30

22.5.1 CAN Control Register (DCAN CTL)

The CAN control register (DCAN CTL) is shown in Figure 22-19 and described in Table 22-14.

Figure 22-19. CAN Control Register (DCAN CTL)

31–26	25	24	23–21	20	19	18	17	16
Reserved	WUBA	PDR	Reserved	DE3	DE2	DE1	IE1	InitDbg
R-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0

15	14	13–10	9	8	7	6	5	4	3	2	1	0
SWR	Reserved	PMD	ABO	IDS	Test	CCE	DAR	Reserved	EIE	SIE	IE0	Init
R/WP-0	R-0	R/W-0x5	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-1

LEGEND: R/W = Read/Write; R = Read only; WP = Write protected by init bit; *-n* = value after reset

Table 22-14. CAN Control Register (DCAN CTL) Field Descriptions

Bit	Field	Value	Description
31-26	Reserved	0	These bits are always read as 0. Writes have no effect.
26	WUBA		Automatic wake up on bus activity when in local power-down mode
		0	No detection of a dominant CAN bus level while in local power-down mode.
		1	Detection of a dominant CAN bus level while in local power-down mode is enabled. On occurrence of a dominant CAN bus level, the wake up sequence is started (Additional information can be found in *Local Power-Down Mode*).
			Note: The CAN message, which initiates the bus activity, cannot be received. This means that the first message received in power down and automatic wake-up mode, will be lost.
24	PDR		Request for local low power-down mode
		0	No application request for local low power-down mode. If the application has cleared this bit while DCAN in local power-down mode, also the Init bit has to be cleared.
		1	Local power-down mode has been requested by application. The DCAN will acknowledge the local power-down mode by setting bit PDA in the error and status register. The local clocks will be turned off by DCAN internal logic (Additional information can be found in *Local Power-Down Mode*).
23-21	Reserved	0	These bits are always read as 0. Writes have no effect.
20	DE3		Enable DMA request line for IF3
		0	Disabled
		1	Enabled
			Note: A pending DMA request for IF3 remains active until first access to one of the IF3 registers.
19	DE2		Enable DMA request line for IF2
		0	Disabled
		1	Enabled
			Note: A pending DMA request for IF2 remains active until first access to one of the IF2 registers.
18	DE1		Enable DMA request line for IF1
		0	Disabled
		1	Enabled
			Note: A pending DMA request for IF1 remains active until first access to one of the IF1 registers.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 22-14. CAN Control Register (DCAN CTL) Field Descriptions (continued)

Bit	Field	Value	Description
17	IE1		Interrupt line 1 enable
		0	Disabled - Module interrupt DCAN1INT is always low.
		1	Enabled - interrupts will assert line DCAN1INT to one; line remains active until pending interrupts are processed.
16	InitDbg		Internal init state while debug access
		0	Not in debug mode, or debug mode requested but not entered.
		1	Debug mode requested and internally entered; the DCAN is ready for debug accesses.
15	SWR		SW reset enable
		0	Normal Operation
		1	Module is forced to reset state. This bit will automatically get cleared after execution of SW reset after one OCP clock cycle. Note: To execute software reset, the following procedure is necessary: 1. Set Init bit to shut down CAN communication. 2. Set SWR bit additionally to Init bit.
14	Reserved	0	This bit is always read as 0. Writes have no effect.
13-10	PMD		Parity on/off
		0x5	Parity function disabled
		Others	Parity function enabled
9	ABO		Auto-Bus-On enable
		0	The Auto-Bus-On feature is disabled
		1	The Auto-Bus-On feature is enabled
8	IDS		Interruption debug support enable
		0	When Debug/Suspend mode is requested, DCAN will wait for a started transmission or reception to be completed before entering Debug/Suspend mode
		1	When Debug/Suspend mode is requested, DCAN will interrupt any transmission or reception, and enter Debug/Suspend mode immediately.
7	Test		Test mode enable
		0	Normal Operation
		1	Test Mode
6	CCE		Configuration change enable
		0	The CPU has no write access to the configuration registers.
		1	The CPU has write access to the configuration registers (when Init bit is set).
5	DAR		Disable automatic retransmission
		0	Automatic retransmission of not successful messages enabled.
		1	Automatic retransmission disabled.
4	Reserved	0	This bit is always read as 0. Writes have no effect.
3	EIE		Error interrupt enable
		0	Disabled - PER, BOff and EWarn bits can not generate an interrupt.
		1	Enabled - PER, BOff and EWarn bits can generate an interrupt at DCAN0INT line and affect the interrupt register.
2	SIE		Status change interrupt enable
		0	Disabled - WakeUpPnd, RxOk, TxOk and LEC bits can not generate an interrupt.
		1	Enabled - WakeUpPnd, RxOk, TxOk and LEC can generate an interrupt at DCAN0INT line and affect the interrupt register.
1	IE0		Interrupt line 0 enable
		0	Disabled - Module interrupt DCAN0INT is always low.
		1	Enabled - interrupts will assert line DCAN0INT to one; line remains active until pending interrupts are processed.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 22-14. CAN Control Register (DCAN CTL) Field Descriptions (continued)

Bit	Field	Value	Description
0	Init		Initialization
		0	Normal operation
		1	Initialization mode is entered

NOTE: The Bus-Off recovery sequence (refer to CAN specification) cannot be shortened by setting or resetting Init bit. If the module goes Bus-Off, it will automatically set the Init bit and stop all bus activities.

When the Init bit is cleared by the application again, the module will then wait for 129 occurrences of Bus Idle (129 * 11 consecutive recessive bits) before resuming normal operation. At the end of the bus-off recovery sequence, the error counters will be reset.

After the Init bit is reset, each time when a sequence of 11 recessive bits is monitored, a Bit0 error code is written to the error and status register, enabling the CPU to check whether the CAN bus is stuck at dominant or continuously disturbed, and to monitor the proceeding of the bus-off recovery sequence.

22.5.2 Error and Status Register (DCAN ES) and DCAN Parity Error EOI Register (PARITYERR_EOI)

The error and status register (DCAN ES) is shown in Figure 22-20 and described in Table 22-15. The DCAN Parity Error EOI register is shown in Figure 22-21 and described in Table 22-16.

Figure 22-20. Error and Status Register (DCAN ES)

31–16	15–11	10	9	8	7	6	5	4	3	2–0
Reserved	Reserved	PDA	Wake UpPnd	PER	BOff	EWarn	EPass	RxOK	TxOK	LEC
R-0	R-0	R-0	R/C-0	R/C-0	R-0	R-0	R-0	R/C-0	R/C-0	R/S-111

LEGEND: R = Read only; S = Set by Read; C = Clear by Read; -*n* = value after reset

Figure 22-21. Parity Error End of Interrupt Register (PARITYERR_EOI)

31–16	15–9	8	0
Reserved	Reserved	PARITYERR_EOI	Reserved
R-0	R-0		R-0

LEGEND: R = Read only; S = Set by Read; C = Clear by Read; -*n* = value after reset

Table 22-15. Error and Status Register (DCAN ES) Field Descriptions

Bit	Field	Value	Description
31-11	Reserved	0	These bits are always read as 0. Writes have no effect.
10	PDA		Local power-down mode acknowledge
		0	DCAN is not in local power-down mode.
		1	Application request for setting DCAN to local power-down mode was successful. DCAN is in local power-down mode.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 22-15. Error and Status Register (DCAN ES) Field Descriptions (continued)

Bit	Field	Value	Description
9	WakeUp Pnd		Wake up pending. This bit can be used by the CPU to identify the DCAN as the source to wake up the system.
		0	No Wake Up is requested by DCAN.
		1	DCAN has initiated a wake up of the system due to dominant CAN bus while module power down.
			This bit will be reset if error and status register is read.
8	PER		Parity error detected
		0	No parity error has been detected since last read access.
		1	The parity check mechanism has detected a parity error in the Message RAM.
			This bit will be reset if error and status register is read.
7	BOff		Bus-Off state
		0	The CAN module is not bus-off state.
		1	The CAN module is in bus-off state.
6	EWarn		Warning state
		0	Both error counters are below the error warning limit of 96.
		1	At least one of the error counters has reached the error warning limit of 96.
5	EPass		Error passive state
		0	On CAN Bus error, the DCAN could send active error frames.
		1	The CAN core is in the error passive state as defined in the CAN Specification.
4	RxOk		Received a message successfully
		0	No message has been successfully received since the last time when this bit was read by the CPU. This bit is never reset by DCAN internal events.
		1	A message has been successfully received since the last time when this bit was reset by a read access of the CPU (independent of the result of acceptance filtering).
			This bit will be reset if error and status register is read.
3	TxOk		Transmitted a message successfully
		0	No message has been successfully transmitted since the last time when this bit was read by the CPU. This bit is never reset by DCAN internal events.
		1	A message has been successfully transmitted (error free and acknowledged by at least one other node) since the last time when this bit was reset by a read access of the CPU.
			This bit will be reset if error and status register is read.
2-0	LEC		Last error code. The LEC field indicates the type of the last error on the CAN bus. This field will be cleared to '0' when a message has been transferred (reception or transmission) without error.
		0	No error
		1	Stuff error: More than five equal bits in a row have been detected in a part of a received message where this is not allowed.
		2	Form error: A fixed format part of a received frame has the wrong format.
		3	Ack error: The message this CAN core transmitted was not acknowledged by another node.
		4	Bit1 error: During the transmission of a message (with the exception of the arbitration field), the device wanted to send a recessive level (bit of logical value '1'), but the monitored bus value was dominant.
		5	Bit0 error: During the transmission of a message (or acknowledge bit, or active error flag, or overload flag), the device wanted to send a dominant level (logical value '0'), but the monitored bus level was recessive. During Bus-Off recovery, this status is set each time a sequence of 11 recessive bits has been monitored. This enables the CPU to monitor the proceeding of the Bus-Off recovery sequence (indicating the bus is not stuck at dominant or continuously disturbed).
		6	CRC error: In a received message, the CRC check sum was incorrect. (CRC received for an incoming message does not match the calculated CRC for the received data).
		7	No CAN bus event was detected since the last time the CPU read the error and status register. Any read access to the error and status register re-initializes the LEC to value '7.'

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 22-16. Parity Error End of Interrupt Register (PARITYERR_EOI) Field Descriptions

Bit	Field	Value	Description
31-9	Reserved	0	These bits are always read as 0. Writes have no effect.
8	PARITYERR_ EOI		End of interrupt indication for parity error
		0	Has no effect writing
		1	Indicated end of interrupt for parity error
7-0	Reserved		These bits are always read as 0. Writes have no effect.

Interrupts are generated by bits PER, BOff and EWarn (if EIE bit in CAN control register is set) and by bits WakeUpPnd, RxOk, TxOk, and LEC (if SIE bit in CAN control register is set).

A change of bit EPass will not generate an interrupt.

PARITYERR_EOI is used to signal an end of interrupt of the parity error interrupt (dcan_parity) to the IP. PARITYERR_EOI register is a *Write only* register.

A read from this offset address will result in a read from the DCAN ES register, where as a write will result in a write to the PARITYERR_EOI register.

NOTE: Reading the error and status register clears the WakeUpPnd, PER, RxOk and TxOk bits and set the LEC to value '7.' Additionally, the status interrupt value (0x8000) in the interrupt register will be replaced by the next lower priority interrupt value. The EOI for all other interrupts (DCANINT0 and DCANINT1) are automatically handled by hardware.

NOTE: For debug support, the auto clear functionality of error and status register (clear of status flags by read) is disabled when in debug/suspend mode.

22.5.3 Error Counter Register (DCAN ERRC)

The error counter register (DCAN ERRC) is shown in Figure 22-22 and described in Table 22-17.

Figure 22-22. Error Counter Register (DCAN ERRC)

31–16	15	14–8	7–0
Reserved	RP	REC[6:0]	TEC[7:0]
R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 22-17. Error Counter Register (DCAN ERRC) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	These bits are always read as 0. Writes have no effect.
15	RP		Receive error passive
		0	The receive error counter is below the error passive level.
		1	The receive error counter has reached the error passive level as defined in the CAN specification.
14-8	REC[6:0]		Receive error counter. Actual state of the receive error counter (values from 0 to 255).
7-0	TEC[7:0]		Transmit error counter. Actual state of the transmit error counter (values from 0 to 255).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.5.4 Bit Timing Register (DCAN BTR)

The bit timing register (DCAN BTR) is shown in Figure 22-23 and described in Table 22-18.

Figure 22-23. Bit Timing Register (DCAN BTR)

31–20	19–16
Reserved	BRPE
R-0	R-0

15	14–12	11–8	7–6	5–0
Reserved	TSeg2	TSeg1	SJW	BRP
R-0	R/WP-0x2	R/WP-0x3	R/WP-0	R/WP-0x1

LEGEND: R = Read only; WP = Write protected by CCE bit; -*n* = value after reset

Table 22-18. Bit Timing Register (DCAN BTR) Field Descriptions

Bit	Field	Value	Description
31-20	Reserved	0	These bits are always read as 0. Writes have no effect.
19-16	BRPE	0x00-0x0F	Baud rate prescaler extension. Valid programmed values are 0 to 15. By programming BRPE the baud rate prescaler can be extended to values up to 1024.
15	Reserved	0	These bits are always read as 0. Writes have no effect.
14-12	TSeg2	0x0-0x7	Time segment after the sample point Valid programmed values are 0 to 7. The actual TSeg2 value which is interpreted for the bit timing will be the programmed TSeg2 value + 1.
11-8	TSeg1	0x01-0x0F	Time segment before the sample point Valid programmed values are 1 to15. The actual TSeg1 value interpreted for the bit timing will be the programmed TSeg1 value + 1.
7-6	SJW	0x0-0x3	Synchronization Jump Width Valid programmed values are 0 to 3. The actual SJW value interpreted for the synchronization will be the programmed SJW value + 1.
5-0	BRP	0x00-0x3F	Baud rate prescaler Value by which the CAN_CLK frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid programmed values are 0 to 63. The actual BRP value interpreted for the bit timing will be the programmed BRP value + 1.

NOTE: This register is only writable if CCE and Init bits in the CAN control register are set.

NOTE: The CAN bit time may be programmed in the range of 8 to 25 time quanta.

NOTE: The CAN time quantum may be programmed in the range of 1 to1024 CAN_CLK periods.

With a CAN_CLK of 8 MHz and BRPE = 0x00, the reset value of 0x00002301 configures the DCAN for a bit rate of 500kBit/s.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.5.5 Interrupt Register (DCAN INT)

The interrupt register (DCAN INT) is shown in Figure 22-24 and described in Table 22-19.

Figure 22-24. Interrupt Register (DCAN INT)

31–24	23–16
Reserved	Int1ID[7:0]
R-0	R-0

15–0
Int0ID[15:0]
R-0

LEGEND: R = Read only; -*n* = value after reset

Table 22-19. Interrupt Register (DCAN INT) Field Descriptions

Bit	Field	Value	Description
31-24	Reserved	0	These bits are always read as 0. Writes have no effect.
23-16	Int1ID[23:16]		Interrupt 1 Identifier (indicates the message object with the highest pending interrupt)
		0x00	No interrupt is pending
		0x01-0x80	Number of message object which caused the interrupt.
		0x81-0xFF	Unused
			If several interrupts are pending, the CAN interrupt register will point to the pending interrupt with the highest priority. The DCAN1INT interrupt line remains active until Int1ID reaches value 0 (the cause of the interrupt is reset) or until IE1 is cleared.
			A message interrupt is cleared by clearing the message object's IntPnd bit.
			Among the message interrupts, the message object's interrupt priority decreases with increasing message number.
15-0	Int0ID[15:0]		Interrupt Identifier (the number here indicates the source of the interrupt)
		0x0000	No interrupt is pending
		0x0001-0x0080	Number of message object which caused the interrupt.
		0x0081-0x7FFF	Unused
		0x8000	Error and status register value is not 0x07.
		0x8001-0xFFFF	Unused
			If several interrupts are pending, the CAN interrupt register will point to the pending interrupt with the highest priority. The DCAN0INT interrupt line remains active until Int0ID reaches value 0 (the cause of the interrupt is reset) or until IE0 is cleared.
			The Status interrupt has the highest priority. Among the message interrupts, the message object's interrupt priority decreases with increasing message number.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.5.6 Test Register (DCAN TEST)

The test register (DCAN TEST) is shown in Figure 22-25 and described in Table 22-20.

Figure 22-25. Test Register (DCAN TEST)

31											16
Reserved											
R-0											

15	10	9	8	7	6	5	4	3	2	1	0
Reserved		RDA	EXL	Rx	Tx[1:0]		LBack	Silent	Reserved		
R-0		R/WP-0	R/WP-0	R-U	R/WP-0		R/WP-0	R/WP-0	R-0		

LEGEND: R = Read only; WP = Write protected by test bit; -*n* = value after reset; U = Undefined

Table 22-20. Test Register (DCAN TEST) Field Descriptions

Bit	Field	Value	Description
31-10	Reserved	0	These bits are always read as 0. Writes have no effect.
9	RDA		RAM direct access enable
		0	Normal pperation
		1	Direct access to the RAM is enabled while in test mode
8	EXL		External loopback mode
		0	Disabled
		1	Enabled
7	Rx		Receive pin. Monitors the actual value of the CAN_RX pin
		0	The CAN bus is dominant
		1	The CAN bus is recessive
6-5	Tx[1:0]		Control of CAN_TX pin
		00	Normal operation, CAN_TX is controlled by the CAN core.
		01	Sample point can be monitored at CAN_TX pin.
		10	CAN_TX pin drives a dominant value.
		11	CAN_TX pin drives a recessive value.
4	LBack		Loopback mode
		0	Disabled
		1	Enabled
3	Silent		Silent mode
		0	Disabled
		1	Enabled
2-0	Reserved	0	These bits are always read as 0. Writes have no effect.

For all test modes, the test bit in CAN control register needs to be set to one. If test bit is set, the RDA, EXL, Tx1, Tx0, LBack and Silent bits are writable. Bit Rx monitors the state of pin CAN_RX and therefore is only readable. All test register functions are disabled when test bit is cleared.

NOTE: The test register is only writable if test bit in CAN control register is set.

NOTE: Setting Tx[1:0] other than '00' will disturb message transfer.

NOTE: When the internal loop-back mode is active (bit LBack is set), bit EXL will be ignored.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.5.7 Parity Error Code Register (DCAN PERR)

The parity error code register (DCAN PERR) is shown in Figure 22-26 and described in Table 22-21.

Figure 22-26. Parity Error Code Register (DCAN PERR)

31–16	15–11	10–8	7–0
Reserved	Reserved	Word Number	Message Number
R-0	R-0	R-U	R-U

LEGEND: R = Read only; -*n* = value after reset; U = Undefined

Table 22-21. Parity Error Code Register (DCAN PERR) Field Descriptions

Bit	Field	Value	Description
31-11	Reserved	0	These bits are always read as 0. Writes have no effect.
10-8	Word Number	0x01-0x05	Word number where parity error has been detected RDA word number (1 to 5) of the message object (according to the message RAM representation in RDA mode).
7-0	Message Number	0x01-0x80	Message object number where parity error has been detected

If a parity error is detected, the PER flag will be set in the error and status register. This bit is not reset by the parity check mechanism; it must be reset by reading the error and status register.

In addition to the PER flag, the parity error code register will indicate the memory area where the parity error has been detected (message number and word number).

If more than one word with a parity error was detected, the highest word number with a parity error will be displayed.

After a parity error has been detected, the register will hold the last error code until power is removed.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.5.8 Auto-Bus-On Time Register (DCAN ABOTR)

The Auto-Bus-On time register (DCAN ABOTR) is shown in Figure 22-27 and described in Table 22-22.

Figure 22-27. Auto-Bus-On Time Register (DCAN ABOTR)

31 – 16	15 – 0
ABO Time[31:16]	ABO Time[15:0]
R/W-0	R/W-0

LEGEND: R = Read only; W = Write only; -*n* = value after reset

Table 22-22. Auto-Bus-On Time Register (DCAN ABOTR) Field Descriptions

Bit	Field	Value	Description
31-0	ABO Time		Number of OCP clock cycles before a Bus-Off recovery sequence is started by clearing the Init bit. This function has to be enabled by setting bit ABO in CAN control register. The Auto-Bus-On timer is realized by a 32-bit counter that starts to count down to zero when the module goes Bus-Off. The counter will be reloaded with the preload value of the ABO time register after this phase.

NOTE: On write access to the CAN control register while Auto-Bus-On timer is running, the Auto-Bus-On procedure will be aborted.

NOTE: During Debug/Suspend mode, running Auto-Bus-On timer will be paused.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.5.9 Transmission Request X Register (DCAN TXRQ X)

The transmission request X register (DCAN TXRQ X) is shown in Figure 22-28.

Figure 22-28. Transmission Request X Register (DCAN TXRQ X)

31															16
Reserved															
R-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TxRqstReg8		TxRqstReg7		TxRqstReg6		TxRqstReg5		TxRqstReg4		TxRqstReg3		TxRqstReg2		TxRqstReg1	
R-0		R-0		R-0		R-0		R-0		R-0		R-0		R-0	

LEGEND: R = Read only; -*n* = value after reset

Example 1. Bit 0 of the transmission request X register represents byte 0 of the transmission request 1 register. If one or more bits in this byte are set, bit 0 of the transmission request X register will be set.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.5.10 Transmission Request Registers (DCAN TXRQ12 to DCAN TXRQ78)

These registers hold the TxRqst bits of the implemented message objects. By reading out these bits, the CPU can check for pending transmission requests. The TxRqst bit in a specific message object can be set/reset by the CPU via the IF1/IF2 message interface registers, or by the message handler after reception of a remote frame or after a successful transmission.

The transmission request registers are shown in Table 22-23 and described in Table 22-24.

Table 22-23. Transmission Request Registers (DCAN TXRQ12 to DCAN TXRQ78)

Offset/Register	31/15	30/14	29/13	29/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
0x88 CAN TXRQ12	TxRqst[32:17]															
	R-0															
	TxRqst[16:1]															
	R-0															
0x8C CAN TXRQ34	TxRqst[64:49]															
	R-0															
	TxRqst[48:33]															
	R-0															
0x90 CAN TXRQ56	TxRqst[96:81]															
	R-0															
	TxRqst[80:65]															
	R-0															
0x94 CAN TXRQ78	TxRqst[128:113]															
	R-0															
	TxRqst[112:97]															
	R-0															

LEGEND: R = Read only; *-n* = value after reset

Table 22-24. Transmission Request Registers (DCAN TXRQ12 to DCAN TXRQ78) Field Descriptions

Bit	Field	Value	Description
31-0	TxRqs[128:1]		Transmission request bits (for all message objects)
		0	No transmission has been requested for this message object.
		1	The transmission of this message object is requested and is not yet done.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.5.11 New Data X Register (DCAN NWDAT X)

With the new data X register, the CPU can detect if one or more bits in the different new data registers are set. Each register bit represents a group of eight message objects. If at least on of the NewDat bits of these message objects are set, the corresponding bit in the new data X register will be set.

The new data X register (DCAN NWDAT X) is shown in Figure 22-29.

Figure 22-29. Transmission Request X Register (DCAN TXRQ X)

31															16
Reserved															
R-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NewDatReg8		NewDatReg7		NewDatReg6		NewDatReg5		NewDatReg4		NewDatReg3		NewDatReg2		NewDatReg1	
R-0		R-0		R-0		R-0		R-0		R-0		R-0		R-0	

LEGEND: R = Read only; -*n* = value after reset

Example 1. Bit 0 of the new data X register represents byte 0 of the new data 1 register. If one or more bits in this byte are set, bit 0 of the new data X register will be set.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.5.12 New Data Registers (DCAN NWDAT12 to DCAN NWDAT78)

These registers hold the NewDat bits of the implemented message objects. By reading out these bits, the CPU can check for new data in the message objects. The NewDat bit of a specific message object can be set/reset by the CPU via the IF1/IF2 interface register sets, or by the message handler after reception of a data frame or after a successful transmission.

The new data registers (DCAN NWDAT12 to DCAN NWDAT78) are shown in Table 22-25 and described in Table 22-26.

Table 22-25. New Data Registers (DCAN NWDAT12 to DCAN NWDAT78)

Offset/Register	31/15	30/14	29/13	29/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
0x9C CAN NWDAT12	NewDat[32:17]															
	R-0															
	NewDat[16:1]															
	R-0															
0xA0 CAN NWDAT34	NewDat[64:49]															
	R-0															
	NewDat[48:33]															
	R-0															
0xA4 CAN NWDAT56	NewDat[96:81]															
	R-0															
	NewDat[80:65]															
	R-0															
0xA8 CAN NWDAT78	NewDat[128:113]															
	R-0															
	NewDat[112:97]															
	R-0															

LEGEND: R = Read only; -*n* = value after reset

Table 22-26. New Data Registers (DCAN NWDAT12 to DCAN NWDAT78) Field Descriptions

Bit	Field	Value	Description
31-0	NewDat[128:1]		New Data Bits (for all message objects)
		0	No new data has been written into the data portion of this message object by the message handler since the last time when this flag was cleared by the CPU.
		1	The message handler or the CPU has written new data into the data portion of this message object.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.5.13 Interrupt Pending X Register (DCAN INTPND X)

With the interrupt pending X register, the CPU can detect if one or more bits in the different interrupt pending registers are set. Each bit of this register represents a group of eight message objects. If at least one of the IntPnd bits of these message objects are set, the corresponding bit in the interrupt pending X register will be set.

The interrupt pending X register (DCAN INTPND X) is shown in Figure 22-30.

Figure 22-30. Interrupt Pending X Register (DCAN INTPND X)

31															16
Reserved															
R-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IntPndReg8		IntPndReg7		IntPndReg6		IntPndReg5		IntPndReg4		IntPndReg3		IntPndReg2		IntPndReg1	
R-0		R-0		R-0		R-0		R-0		R-0		R-0		R-0	

LEGEND: R = Read only; -*n* = value after reset

Example 2. Bit 0 of the interrupt pending X register represents byte 0 of the interrupt pending 1 register. If one or more bits in this byte are set, bit 0 of the interrupt pending X register will be set.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.5.14 Interrupt Pending Registers (DCAN INTPND12 to DCAN INTPND78)

These registers hold the IntPnd bits of the implemented message objects. By reading out these bits, the CPU can check for pending interrupts in the message objects. The IntPnd bit of a specific message object can be set/reset by the CPU via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission.

The interrupt pending registers (DCAN INTPND12 to DCAN INTPND78) are shown in Table 22-27 and described in Table 22-28.

Table 22-27. New Data Registers (DCAN NWDAT12 to DCAN NWDAT78)

Offset/Register	31/15	30/14	29/13	29/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
0xB0 CAN INTPND12	IntPnd[32:17]															
	R-0															
	IntPnd[16:1]															
	R-0															
0xB4 CAN INTPND34	IntPnd[64:49]															
	R-0															
	IntPnd[48:33]															
	R-0															
0xB8 CAN INTPND56	IntPnd[96:81]															
	R-0															
	IntPnd[80:65]															
	R-0															
0xBC CAN INTPND78	IntPnd[128:113]															
	R-0															
	IntPnd[112:97]															
	R-0															

LEGEND: R = Read only; -*n* = value after reset

Table 22-28. New Data Registers (DCAN NWDAT12 to DCAN NWDAT78) Field Descriptions

Bit	Field	Value	Description
31-0	IntPnd[128:1]		Interrupt Pending Bits (for all message objects)
		0	This message object is not the source of an interrupt.
		1	This message object is the source of an interrupt.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.5.15 Message Valid X Register (DCAN MSGVAL X)

With the message valid X register, the CPU can detect if one or more bits in the different message valid registers are set. Each bit of this register represents a group of eight message objects. If at least one of the MsgVal bits of these message objects are set, the corresponding bit in the message valid X register will be set.

The message valid X register (DCAN MSGVAL X) is shown in Figure 22-31.

Figure 22-31. Message Valid X Register (DCAN MSGVAL X)

31															16
Reserved															
R-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MsgValReg8		MsgValReg7		MsgValReg6		MsgValReg5		MsgValReg4		MsgValReg3		MsgValReg2		MsgValReg1	
R-0		R-0		R-0		R-0		R-0		R-0		R-0		R-0	

LEGEND: R = Read only; -*n* = value after reset

Example 3. Bit 0 of the message valid X register represents byte 0 of the message valid 1 register. If one or more bits in this byte are set, bit 0 of the message valid X register will be set.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.5.16 Message Valid Registers (DCAN MSGVAL12 to DCAN MSGVAL78)

These registers hold the MsgVal bits of the implemented message objects. By reading out these bits, the CPU can check which message objects are valid. The MsgVal bit of a specific message object can be set/reset by the CPU via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission.

The message valid registers (DCAN MSGVAL12 to MSGVAL78) is shown in Table 22-29 and described in Table 22-30.

Table 22-29. Message Valid Registers (DCAN MSGVAL12 to DCAN MSGVAL78)

Offset/Register	31/15	30/14	29/13	29/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
0xC4 CAN MSGVAL12	MsgVal[32:17]															
	R-0															
	MsgVal[16:1]															
	R-0															
0xC8 CAN MSGVAL34	MsgVal[64:49]															
	R-0															
	MsgVal[48:33]															
	R-0															
0xCC CAN MSGVAL56	MsgVal[96:81]															
	R-0															
	MsgVal[80:65]															
	R-0															
0xD0 CAN MSGVAL78	MsgVal[128:113]															
	R-0															
	MsgVal[112:97]															
	R-0															

LEGEND: R = Read only; -*n* = value after reset

Table 22-30. Message Valid Registers (DCAN MSGVAL12 to DCAN MSGVAL78) Field Descriptions

Bit	Field	Value	Description
31-0	MsgVal[128:1]		Message valid bits (for all message objects)
		0	This message object is ignored by the message handler.
		1	This message object is configured and will be considered by the message handler.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.5.17 Interrupt Multiplexer Registers (DCAN INTMUX12 to DCAN INTMUX78)

The IntMux flag determine for each message object, which of the two interrupt lines (DCAN0INT or DCAN1INT) will be asserted when the IntPnd of this message object is set. Both interrupt lines can be globally enabled or disabled by setting or clearing IE0 and IE1 bits in CAN control register.

The IntPnd bit of a specific message object can be set or reset by the CPU via the IF1/IF2 interface register sets, or by message handler after reception or successful transmission of a frame. This will also affect the Int0ID resp Int1ID flags in the interrupt register.

The interrupt multiplexer registers are shown in Table 22-31 and described in Table 22-32.

Table 22-31. Interrupt Multiplexer Registers (DCAN INTMUX12 to DCAN INTMUX78)

Offset/Register	31/15	30/14	29/13	29/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
0xD8 CAN INTMUX12	IntMux[32:17]															
	R-0															
	IntMux[16:1]															
	R-0															
0xDC CAN INTMUX34	IntMux[64:49]															
	R-0															
	IntMux[48:33]															
	R-0															
0xE0 CAN INTMUX56	IntMux[96:81]															
	R-0															
	IntMux[80:65]															
	R-0															
0xE4 CAN INTMUX78	IntMux[128:113]															
	R-0															
	IntMux[112:97]															
	R-0															

LEGEND: R = Read only; -*n* = value after reset

Table 22-32. Interrupt Multiplexer Registers (DCAN INTMUX12 to DCAN INTMUX78) Field Descriptions

Bit	Field	Value	Description
31-0	IntMux[128:1]		Multiplexes IntPnd value to either DCAN0INT or DCAN1INT interrupt lines (for all message objects)
		0	DCAN0INT line is active if corresponding IntPnd flag is one.
		1	DCAN1INT line is active if corresponding IntPnd flag is one.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.5.18 IF1/IF2 Command Registers (DCAN IF1CMD, DCAN IF2CMD)

The IF1/IF2 Command Register (DCAN IF1CMD, DCAN IF2CMD) configure and initiate the transfer between the IF1/IF2 register sets and the message RAM. It is configurable which portions of the message object should be transferred.

A transfer is started when the CPU writes the message number to bits [7:0] of the IF1/IF2 command register. With this write operation, the Busy bit is automatically set to '1' to indicate that a transfer is in progress.

After 4 to 14 OCP clock cycles, the transfer between the interface register and the message RAM will be completed and the Busy bit is cleared. The maximum number of cycles is needed when the message transfer concurs with a CAN message transmission, acceptance filtering, or message storage.

If the CPU writes to both IF1/IF2 command registers consecutively (request of a second transfer while first transfer is still in progress), the second transfer will start after the first one has been completed.

The IF1/IF2 command registers are shown in Figure 22-32 and Figure 22-33 and described in Table 22-33.

NOTE: While Busy bit is one, IF1/IF2 register sets are write protected.

NOTE: For debug support, the auto clear functionality of the IF1/IF2 command registers (clear of DMAactive flag by r/w) is disabled during Debug/Suspend mode.

NOTE: If an invalid Message Number is written to bits [7:0] of the IF1/IF2 command register, the message handler may access an implemented (valid) message object instead.

Figure 22-32. IF1 Command Registers (DCAN IF1CMD)

31–24	23	22	21	20	19	18	17	16
Reserved	WR/RD	Mask	Arb	Control	Clr IntPnd	TxRqst/New-Dat	Data A	Data B
R-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0

15	14	13–8	7–0
Busy	DMA active	Reserved	Message Number
R-0	R/WP/C-0	R-0	R/WP-0x1

LEGEND: R = Read only; WP = Write protected (protected by busy bit); C = Clear by IF1 access; -*n* = value after reset

Figure 22-33. IF2 Command Registers (DCAN IF2CMD)

31–24	23	22	21	20	19	18	17	16
Reserved	WR/RD	Mask	Arb	Control	Clr IntPnd	TxRqst/New-Dat	Data A	Data B
R-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0

15	14	13–8	7–0
Busy	DMA active	Reserved	Message Number
R-0	R/WP/C-0	R-0	R/WP-0x1

LEGEND: R = Read only; WP = Write protected (protected by busy bit); C = Clear by IF2 access; -*n* = value after reset

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 22-33. IF1/IF2 Command Registers (DCAN IF1CMD, DCAN IF2CMD) Field Descriptions

Bit	Field	Value	Description
31-24	Reserved	0	These bits are always read as 0. Writes have no effect.
23	WR/RD		Write/Read
		0	Direction = Read: Transfer direction is from the message object addressed by Message Number (Bits [7:0]) to the IF1/IF2 register set.
		1	Direction = Write: Transfer direction is from the IF1/IF2 register set to the message object addressed by Message Number (Bits [7:0]).
22	Mask		Access mask bits
		0	Mask bits will not be changed
		1	Direction = Read: The mask bits (identifier mask + MDir + MXtd) will be transferred from the message object addressed by Message Number (Bits [7:0]) to the IF1/IF2 register set.
			Direction = Write: The mask bits (identifier mask + MDir + MXtd) will be transferred from the IF1/IF2 register set to the message object addressed by Message Number (Bits [7:0]).
21	Arb		Access arbitration bits
		0	Arbitration bits will not be changed
		1	Direction = Read: The Arbitration bits (Identifier + Dir + Xtd + MsgVal) will be transferred from the message object addressed by Message Number (Bits [7:0]) to the corresponding IF1/IF2 register set.
			Direction = Write: The Arbitration bits (Identifier + Dir + Xtd + MsgVal) will be transferred from the IF1/IF2 register set to the message object addressed by Message Number (Bits [7:0]).
20	Control		Access control bits
		0	Control bits will not be changed
		1	Direction = Read: The message control bits will be transferred from the message object addressed by message number (Bits [7:0]) to the IF1/IF2 registerset.
			Direction = Write: The message control bits will be transferred from the IF1/IF2 registerset to the message object addressed by message number (Bits [7:0]).
			If the TxRqst/NewDat bit in this register(Bit [18]) is set, the TxRqst/ NewDat bit in the IF1/IF2 message control register will be ignored.
19	ClrIntPnd		Clear interrupt pending bit
		0	IntPnd bit will not be changed
		1	Direction = Read: Clears IntPnd bit in the message object.
			Direction = Write: This bit is ignored. Copying of IntPnd flag from IF1/IF2 Registers to message RAM can only be controlled by the control flag (Bit [20]).
18	TxRqst/NewDat		Access transmission request bit
		0	Direction = Read: NewDat bit will not be changed. Direction = Write: TxRqst/NewDat bit will be handled according to the control bit.
		1	Direction = Read: Clears NewDat bit in the message object. Direction = Write: Sets TxRqst/NewDat in message object.
			Note: If a CAN transmission is requested by setting TxRqst/NewDat in this register, the TxRqst/NewDat bits in the message object will be set to one independent of the values in IF1/IF2 message control Register.
			Note: A read access to a message object can be combined with the reset of the control bits IntPnd and NewDat. The values of these bits transferred to the IF1/IF2 message control registeralways reflect the status before resetting them.
17	Data A		Access Data Bytes 0-3
		0	Data Bytes 0-3 will not be changed.
		1	Direction = Read: The data bytes 0-3 will be transferred from the message object addressed by the Message Number (Bits [7:0]) to the corresponding IF1/IF2 registerset.
			Direction = Write: The data bytes 0-3 will be transferred from the IF1/IF2 registerset to the message object addressed by the Message Number (Bits [7:0]).
			Note: The duration of the message transfer is independent of the number of bytes to be transferred.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 22-33. IF1/IF2 Command Registers (DCAN IF1CMD, DCAN IF2CMD) Field Descriptions (continued)

Bit	Field	Value	Description
16	Data B		Access Data Bytes 4-7
		0	Data Bytes 4-7 will not be changed.
		1	Direction = Read: The data bytes 4-7 will be transferred from the message object addressed by Message Number (Bits [7:0]) to the corresponding IF1/IF2 registerset. Direction = Write: The data bytes 4-7 will be transferred from the IF1/IF2 registerset to the message object addressed by message number (Bits [7:0]). Note: The duration of the message transfer is independent of the number of bytes to be transferred.
15	Busy		Busy flag
		0	No transfer between IF1/IF2 register set and message RAM is in progress.
		1	Transfer between IF1/IF2 register set and message RAM is in progress. This bit is set to one after the message number has been written to bits [7:0]. IF1/IF2 register set will be write protected. The bit is cleared after read/write action has been finished.
14	DMAactive		Activation of DMA feature for subsequent internal IF1/IF2 update
		0	DMA request line is independent of IF1/IF2 activities.
		1	DMA is requested after completed transfer between IF1/IF2 register set and message RAM. The DMA request remains active until the first read or write to one of the IF1/IF2 registers; an exception is a write to Message Number (Bits [7:0]) when DMAactive is one. Note: Due to the auto reset feature of the DMAactive bit, this bit has to be set for each subsequent DMA cycle separately.
13-8	Reserved	0	These bits are always read as 0. Writes have no effect.
7-0	Message Number		Number of message object in message RAM which is used for data transfer
		0x00	Invalid message number
		0x01-0x80	Valid message numbers
		0x81-0xFF	Invalid message numbers

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.5.19 IF1/IF2 Mask Registers (DCAN IF1MSK, DCAN IF2MSK)

The bits of the IF1/IF2 mask registers mirror the mask bits of a message object. The function of the relevant message objects bits is described in *Structure of Message Objects*.

The IF1/IF2 mask registers (DCAN IF1MSK, DCAN IF2MSK) are shown in Figure 22-34 and Figure 22-35 and described in Table 22-34.

NOTE: While Busy bit of IF1/IF2 command register is one, IF1/IF2 register set is write protected.

Figure 22-34. IF1 Mask Register (DCAN IF1MSK)

31	30	29	28 … 16
MXtd	MDir	Reserved	Msk[28:16]
R/WP-1	R/WP-1	R-1	R/WP-0x1FFF

15 … 0
Msk[15:0]
R/WP-0xFFFF

LEGEND: R = Read only; WP = Write protected (protected by busy bit); -*n* = value after reset

Figure 22-35. IF2 Mask Register (DCAN IF2MSK)

31	30	29	28 … 16
MXtd	MDir	Reserved	Msk[28:16]
R/WP-1	R/WP-1	R-1	R/WP-0x1FFF

15 … 0
Msk[15:0]
R/WP-0xFFFF

LEGEND: R = Read only; WP = Write protected (protected by busy bit); -*n* = value after reset

Table 22-34. IF1/IF2 Mask Registers (DCAN IF1MSK, DCAN IF2MSK) Field Descriptions

Bit	Field	Value	Description
31	MXtd		Mask Extended Identifier
		0	The extended identifier bit (IDE) has no effect on the acceptance filtering.
		1	The extended identifier bit (IDE) is used for acceptance filtering.
			When 11-bit ("standard") identifiers are used for a message object, the identifiers of received data frames are written into bits ID[28:18]. For acceptance filtering, only these bits together with mask bits Msk[28:18] are considered.
30	MDir		Mask Message Direction
		0	The message direction bit (Dir) has no effect on the acceptance filtering.
		1	The message direction bit (Dir) is used for acceptance filtering.
29	Reserved	0	These bits are always read as 0. Writes have no effect.
28-0	Msk[28:0]		Identifier Mask
		0	The corresponding bit in the identifier of the message object is not used for acceptance filtering (don't care).
		1	The corresponding bit in the identifier of the message object is used for acceptance filtering.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.5.20 IF1/IF2 Arbitration Registers (DCAN IF1ARB, DCAN IF2ARB)

The bits of the IF1/IF2 arbitration registers mirror the arbitration bits of a message object. The function of the relevant message objects bits is described in *Structure of Message Objects*.

The IF1/IF2 arbitration registers (DCAN IF1ARB, DCAN IF2ARB) are shown in Figure 22-36 and Figure 22-37 and described in Table 22-35.

NOTE: While Busy bit of IF1/IF2 command register is one, IF1/IF2 register set is write protected.

Figure 22-36. IF1 Arbitration Register (DCAN IF1ARB)

31	30	29	28 … 16
MsgVal	Xtd	Dir	ID[28:16]
R/WP-0	R/WP-0	R/WP-0	R/WP-0

15 … 0
ID[15:0]
R/WP-0

LEGEND: R = Read only; WP = Write protected (protected by busy bit); -*n* = value after reset

Figure 22-37. IF2 Arbitration Register (DCAN IF2ARB)

31	30	29	28 … 16
MsgVal	Xtd	Dir	ID[28:16]
R/WP-0	R/WP-0	R/WP-0	R/WP-0

15 … 0
ID[15:0]
R/WP-0

LEGEND: R = Read only; WP = Write protected (protected by busy bit); -*n* = value after reset

Table 22-35. IF1/IF2 Arbitration Registers (DCAN IF1ARB, DCAN IF2ARB) Field Descriptions

Bit	Field	Value	Description
31	MsgVal		Message valid
		0	The message object is ignored by the message handler.
		1	The message object is to be used by the message handler.
			The CPU should reset the MsgVal bit of all unused Messages Objects during the initialization before it resets bit Init in the CAN control register. This bit must also be reset before the identifier ID[28:0], the control bits Xtd, Dir or DLC[3:0] are modified, or if the messages object is no longer required.
30	Xtd		Extended identifier
		0	The 11-bit (“standard”) Identifier is used for this message object.
		1	The 29-bit (“extended”) Identifier is used for this message object.
29	Dir		Message direction
		0	Direction = receive: On TxRqst, a remote frame with the identifier of this message object is transmitted. On reception of a data frame with matching identifier, this message is stored in this message object.
		1	Direction = transmit: On TxRqst, the respective message object is transmitted as a data frame. On reception of a remote frame with matching identifier, the TxRqst bit of this message object is set (if RmtEn = 1).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 22-35. IF1/IF2 Arbitration Registers (DCAN IF1ARB, DCAN IF2ARB) Field Descriptions (continued)

Bit	Field	Value	Description
28-0	Msk[28:0]		Message identifier
		ID[28:0]	29-bit identifier (extended frame)
		ID[28:18]	11-bit identifier (standard frame)

The Arbitration bits ID[28:0], Xtd, and Dir are used to define the identifier and type of outgoing messages and (together with the mask bits Msk[28:0], MXtd, and MDir) for acceptance filtering of incoming messages.

A received message is stored into the valid message object with matching identifier and Direction = receive (data frame) or Direction = transmit (remote frame).

Extended frames can be stored only in message objects with Xtd = one, standard frames in message objects with Xtd = zero.

If a received message (data frame or remote frame) matches more than one valid message objects, it is stored into the one with the lowest message number.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.5.21 IF1/IF2 Message Control Registers (DCAN IF1MCTL, DCAN IF2MCTL)

The bits of the IF1/IF2 message control registers mirror the message control bits of a message object. The function of the relevant message objects bits is described in *Structure of Message Objects*.

The IF1/IF2 message control registers (DCAN IF1MCTL, DCAN IF2MCTL) are shown in Figure 22-38 and Figure 22-39 and described in Table 22-36.

NOTE: While Busy bit of IF1/IF2 command register is one, IF1/IF2 register set is write protected.

Figure 22-38. IF1 Message Control Register (DCAN IF1MCTL)

31															16
Reserved															
R/WP- 0															
15	14	13	12	11	10	9	8	7	6		4	3	2	1	0
New Dat	Msg Lst	Int Pnd	UMask	TxIE	RxIE	Rmt En	Tx Rqst	EoB	Reserved			DLC[3:0]			
R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0			R/WP-0			

LEGEND: R = Read only; WP = Protected Write (protected by Busy bit); -*n* = value after reset

Figure 22-39. IF2 Message Control Register (DCAN IF2MCTL)

31															16
Reserved															
R/WP- 0															
15	14	13	12	11	10	9	8	7	6		4	3	2	1	0
New Dat	Msg Lst	Int Pnd	UMask	TxIE	RxIE	Rmt En	Tx Rqst	EoB	Reserved			DLC[3:0]			
R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0	R/WP-0			R/WP-0			

LEGEND: R = Read only; WP = Protected Write (protected by Busy bit); -*n* = value after reset

Table 22-36. IF1/IF2 Message Control Registers (DCAN IF1MCTL, DCAN IF2MCTL) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	These bits are always read as 0. Writes have no effect.
15	NewDat		New data
		0	No new data has been written into the data portion of this message object by the message handler since the last time when this flag was cleared by the CPU.
		1	The message handler or the CPU has written new data into the data portion of this message object.
14	MsgLst		Message lost (only valid for message objects with direction = receive)
		0	No message lost since the last time when this bit was reset by the CPU.
		1	The message handler stored a new message into this object when NewDat was still set, so the previous message has been overwritten.
13	IntPnd		Interrupt pending
		0	This message object is not the source of an interrupt.
		1	This message object is the source of an interrupt. The Interrupt Identifier in the interrupt register will point to this message object if there is no other interrupt source with higher priority.
12	UMask		Use acceptance mask
		0	Mask ignored
		1	Use mask (Msk[28:0], MXtd, and MDir) for acceptance filtering
			If the UMask bit is set to one, the message object's mask bits have to be programmed during initialization of the message object before MsgVal is set to one.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 22-36. IF1/IF2 Message Control Registers (DCAN IF1MCTL, DCAN IF2MCTL) Field Descriptions (continued)

Bit	Field	Value	Description
11	TxIE		Transmit interrupt enable
		0	IntPnd will not be triggered after the successful transmission of a frame.
		1	IntPnd will be triggered after the successful transmission of a frame.
10	RxIE		Receive interrupt enable
		0	IntPnd will not be triggered after the successful reception of a frame.
		1	IntPnd will be triggered after the successful reception of a frame.
9	RmtEn		Remote enable
		0	At the reception of a remote frame, TxRqst is not changed.
		1	At the reception of a remote frame, TxRqst is set.
8	TxRqst		Transmit request
		0	This message object is not waiting for a transmission.
		1	The transmission of this message object is requested and is not yet done.
7	EoB		Data frame has 0-8 data bits.
		0	Data frame has 8 data bytes.
		1	Note: The data length code of a message object must be defined the same as in all the corresponding objects with the same identifier at other nodes. When the message handler stores a data frame, it will write the DLC to the value given by the received message.
			Note: This bit is used to concatenate multiple message objects to build a FIFO Buffer. For single message objects (not belonging to a FIFO Buffer), this bit must always be set to one.
6-4	Reserved	0	These bits are always read as 0. Writes have no effect.
3-0			Data length code
		0-8	Data frame has 0-8 data bits.
		9-15	Data frame has 8 data bytes.
			Note: The data length code of a message object must be defined the same as in all the corresponding objects with the same identifier at other nodes. When the message handler stores a data frame, it will write the DLC to the value given by the received message.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.5.22 IF1/IF2 Data A and Data B Registers (DCAN IF1DATA/DATB, DCAN IF2DATA/DATB)

The data bytes of CAN messages are stored in the IF1/IF2 registers in the following order:

1. In a CAN data frame, Data 0 is the first, and Data 7 is the last byte to be transmitted or received.
2. In CAN's serial bit stream, the MSB of each byte will be transmitted first.

The IF1/IF2 data A and data B registers (DCAN IF1DATA/DATB, DCAN IF2DATA/DATB) registers are shown in Figure 22-40, Figure 22-41, Figure 22-42 and Figure 22-43.

Figure 22-40. IF1 Data A Register (DCAN IF1DATA)

31–24	23–16
Data 3	Data 2
R/WP-0	R/WP-0

15–8	7–0
Data 1	Data 0
R/WP-0	R/WP-0

LEGEND: R = Read only; WP = Protected Write (protected by Busy bit); -*n* = value after reset

Figure 22-41. IF1 Data B Register (DCAN IF1DATA)

31–24	23–16
Data 7	Data 6
R/WP-0	R/WP-0

15–8	7–0
Data 5	Data 4
R/WP-0	R/WP-0

LEGEND: R = Read only; WP = Protected Write (protected by Busy bit); -*n* = value after reset

Figure 22-42. IF2 Data A Register (DCAN IF2DATA)

31–24	23–16
Data 7	Data 6
R/WP-0	R/WP-0

15–8	7–0
Data 5	Data 4
R/WP-0	R/WP-0

LEGEND: R = Read only; WP = Protected Write (protected by Busy bit); -*n* = value after reset

Figure 22-43. IF2 Data B Register (DCAN IF2DATA)

31–24	23–16
Data 7	Data 6
R/WP-0	R/WP-0

15–8	7–0
Data 5	Data 4
R/WP-0	R/WP-0

LEGEND: R = Read only; WP = Protected Write (protected by Busy bit); -*n* = value after reset

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.5.23 IF3 Observation Register (DCAN IF3OBS)

The IF3 register set can automatically be updated with received message objects without the need to initiate the transfer from message RAM by CPU (Additional information can be found in *Structure of Message Objects*).

The observation flags (Bits [4:0]) in the IF3 observation register are used to determine, which data sections of the IF3 interface register set have to be read in order to complete a DMA read cycle. After all marked data sections are read, the DCAN is enabled to update the IF3 interface register set with new data.

Any access order of single bytes or half-words is supported. When using byte or half-word accesses, a data section is marked as completed, if all bytes are read.

NOTE: If IF3 Update Enable is used and no Observation flag is set, the corresponding message objects will be copied to IF3 without activating the DMA request line and without waiting for DMA read accesses.

A write access to this register aborts a pending DMA cycle by resetting the DMA line and enables updating of IF3 interface register set with new data. To avoid data inconsistency, the DMA controller should be disabled before reconfiguring IF3 observation register.

The status of the current read-cycle can be observed via status flags (Bits [12:8]).

If an interrupt line is available for IF3, an interrupt will be generated by IF3Upd flag. See the device-specific data sheet for the availability of this interrupt source.

With this interrupt, the observation status bits and the IF3Upd bit could be used by the application to realize the notification about new IF3 content in polling or interrupt mode.

The IF3 observation register (DCAN IF3OBS) is shown in Figure 22-44 and described in Table 22-37.

Figure 22-44. IF3 Observation Register (DCAN IF3OBS)

31														16
Reserved														
R-0														
15	14	13	12	11	10	9	8	7	5	4	3	2	1	0
IF3 Upd	Reserved		IF3 SDB	IF3 SDA	IF3 SC	IF3 SA	IF3 SM	Reserved		DataB	DataA	Ctrl	Arb	Mask
R-0	R-0		R-0	R-0	R-0	R-0	R-0	R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 22-37. IF3 Observation Register (DCAN IF3OBS) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	These bits are always read as 0. Writes have no effect.
15	IF3 Upd		IF3 Update Data
		0	No new data has been loaded since last IF3 read.
		1	New data has been loaded since last IF3 read.
14-13	Reserved	0	These bits are always read as 0. Writes have no effect.
12	IF3 SDB		IF3 Status of Data B read access
		0	All Data B bytes are already read out, or are not marked to be read.
		1	Data B section has still data to be read out.
11	IF3 SDA		IF3 Status of Data A read access
		0	All Data A bytes are already read out, or are not marked to be read.
		1	Data A section has still data to be read out.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 22-37. IF3 Observation Register (DCAN IF3OBS) Field Descriptions (continued)

Bit	Field	Value	Description
10	IF3 SC		IF3 Status of control bits read access
		0	All control section bytes are already read out, or are not marked to be read.
		1	Control section has still data to be read out.
9	IF3 SA		IF3 Status of Arbitration data read access
		0	All Arbitration data bytes are already read out, or are not marked to be read.
		1	Arbitration section has still data to be read out.
8	IF3 SM		IF3 Status of Mask data read access
		0	All mask data bytes are already read out, or are not marked to be read.
		1	Mask section has still data to be read out.
7-5	Reserved	0	These bits are always read as 0. Writes have no effect.
4	DataB		Data B read observation
		0	Data B section has not to be read.
		1	Data B section has to be read to enable next IF3 update.
3	DataA		Data A read observation
		0	Data A section has not to be read.
		1	Data A section has to be read to enable next IF3 update.
2	Ctrl		Ctrl read observation
		0	Ctrl section has not to be read.
		1	Ctrl section has to be read to enable next IF3 update.
1	Arb		Arbitration data read observation
		0	Arbitration data has not to be read.
		1	Arbitration data has to be read to enable next IF3 update.
0	Mask		Mask data read observation
		0	Mask data has not to be read.
		1	Mask data has to be read to enable next IF3 update.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.5.24 IF3 Mask Register (DCAN IF3MSK)

The IF3 mask register (DCAN IF3MSK) is shown in Figure 22-45 and described in Table 22-38.

Figure 22-45. IF3 Mask Register (DCAN IF3MSK)

31	30	29	28 … 16
MXtd	MDir	Reserved	Msk[28:16]
R-1	R-1	R-1	R/WP-0x1FFF

15 … 0
Msk[15:0]
R/WP-0xFFFF

LEGEND: R = Read only; -*n* = value after reset

Table 22-38. IF3 Mask Register (DCAN IF3MSK) Field Descriptions

Bit	Field	Value	Description
31	MXtd		Mask Extended Identifier
		0	The extended identifier bit (IDE) has no effect on the acceptance filtering.
		1	The extended identifier bit (IDE) is used for acceptance filtering.
			When 11-bit (“standard”) identifiers are used for a message object, the identifiers of received data frames are written into bits ID[28:18]. For acceptance filtering, only these bits together with mask bits Msk[28:18] are considered.
30	MDir		Mask Message Direction
		0	The message direction bit (Dir) has no effect on the acceptance filtering.
		1	The message direction bit (Dir) is used for acceptance filtering.
29	Reserved	0	These bits are always read as 0. Writes have no effect.
28-0	Msk[28:0]		Identifier Mask
		0	The corresponding bit in the identifier of the message object is not used for acceptance filtering (don't care).
		1	The corresponding bit in the identifier of the message object is used for acceptance filtering.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.5.25 IF3 Arbitration Register (DCAN IF3ARB)

The IF3 arbitration register (DCAN IF3ARB) is shown in Figure 22-46 and described in Table 22-39.

Figure 22-46. IF3 Arbitration Register (DCAN IF3ARB)

31	30	29	28 … 16
MsgVal	Xtd	Dir	ID[28:16]
R-0	R-0	R-0	R-0

15 … 0
ID[15:0]
R-0

LEGEND: R = Read only; WP = Write protected (protected by busy bit); -*n* = value after reset

Table 22-39. IF3 Arbitration Register (DCAN IF3ARB) Field Descriptions

Bit	Field	Value	Description
31	MsgVal		Message Valid
		0	The message object is ignored by the message handler.
		1	The message object is to be used by the message handler.
			The CPU should reset the MsgVal bit of all unused Messages Objects during the initialization before it resets bit Init in the CAN control register. This bit must also be reset before the identifier ID[28:0], the control bits Xtd, Dir or DLC[3:0] are modified, or if the messages object is no longer required.
30	Xtd		Extended Identifier
		0	The 11-bit (“standard”) Identifier is used for this message object.
		1	The 29-bit (“extended”) Identifier is used for this message object.
29	Dir		Message Direction
		0	Direction = receive: On TxRqst, a remote frame with the identifier of this message object is transmitted. On reception of a data frame with matching identifier, this message is stored in this message object.
		1	Direction = transmit: On TxRqst, the respective message object is transmitted as a data frame. On reception of a remote frame with matching identifier, the TxRqst bit of this message object is set (if RmtEn = 1).
28-0	Msk[28:0]		Message Identifier
		ID[28:0]	29-bit Identifier (“extended frame”)
		ID[28:18]	11-bit Identifier (“standard frame”)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.5.26 IF3 Message Control Register (DCAN IF3MCTL)

The IF3 message control register (DCAN IF3MCTL) is shown in Figure 22-47 and described in Table 22-40.

Figure 22-47. IF3 Message Control Register (DCAN IF3MCTL)

31															16
Reserved															
R-0															
15	14	13	12	11	10	9	8	7	6		4	3	2	1	0
New Dat	Msg Lst	Int Pnd	UMask	TxIE	RxIE	Rmt En	Tx Rqst	EoB	Reserved			DLC[3:0]			
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0			R-0			

LEGEND: R = Read only; -*n* = value after reset

Table 22-40. IF3 Message Control Register (DCAN IF3MCTL) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	These bits are always read as 0. Writes have no effect.
15	NewDat		New Data
		0	No new data has been written into the data portion of this message object by the message handler since the last time when this flag was cleared by the CPU.
		1	The message handler or the CPU has written new data into the data portion of this message object.
14	MsgLst		Message Lost (only valid for message objects with direction = receive)
		0	No message lost since the last time when this bit was reset by the CPU.
		1	The message handler stored a new message into this object when NewDat was still set, so the previous message has been overwritten.
13	IntPnd		Interrupt Pending
		0	This message object is not the source of an interrupt.
		1	This message object is the source of an interrupt. The Interrupt Identifier in the interrupt register will point to this message object if there is no other interrupt source with higher priority.
12	UMask		Use Acceptance Mask
		0	Mask ignored
		1	Use mask (Msk[28:0], MXtd, and MDir) for acceptance filtering
			If the UMask bit is set to one, the message object's mask bits have to be programmed during initialization of the message object before MsgVal is set to one.
11	TxIE		Transmit Interrupt enable
		0	IntPnd will not be triggered after the successful transmission of a frame.
		1	IntPnd will be triggered after the successful transmission of a frame.
10	RxIE		Receive Interrupt enable
		0	IntPnd will not be triggered after the successful reception of a frame.
		1	IntPnd will be triggered after the successful reception of a frame.
9	RmtEn		Remote enable
		0	At the reception of a remote frame, TxRqst is not changed.
		1	At the reception of a remote frame, TxRqst is set.
8	TxRqst		Transmit Request
		0	This message object is not waiting for a transmission.
		1	The transmission of this message object is requested and is not yet done.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 22-40. IF3 Message Control Register (DCAN IF3MCTL) Field Descriptions (continued)

Bit	Field	Value	Description
7	EoB		Data frame has 0-8 data bits.
		0	Data frame has 8 data bytes.
		1	Note: The data length code of a message object must be defined the same as in all the corresponding objects with the same identifier at other nodes. When the message handler stores a data frame, it will write the DLC to the value given by the received message.
			Note: This bit is used to concatenate multiple message objects to build a FIFO Buffer. For single message objects (not belonging to a FIFO Buffer), this bit must always be set to one.
6-4	Reserved	0	These bits are always read as 0. Writes have no effect.
3-0			Data Length Code
		0-8	Data frame has 0-8 data bits.
		9-15	Data frame has 8 data bytes.
			Note: The data length code of a message object must be defined the same as in all the corresponding objects with the same identifier at other nodes. When the message handler stores a data frame, it will write the DLC to the value given by the received message.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.5.27 IF3 Data A and Data B Registers (DCAN IF3DATA/DATB)

The data bytes of CAN messages are stored in the IF3 registers in the following order.

In a CAN data frame, Data 0 is the first, and Data 7 is the last byte to be transmitted or received. In CAN's serial bit stream, the MSB of each byte will be transmitted first.

The IF3 data A and data B registers (DCAN IF3DATA/DATB) are shown in Figure 22-48 and Figure 22-49.

Figure 22-48. IF3 Data A Register (DCAN IF3DATA)

31–24	23–16
Data 3	Data 2
R-0	R-0

15–8	7–0
Data 1	Data 0
R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Figure 22-49. IF3 Data A Register (DCAN IF3DATB)

31–24	23–16
Data 7	Data 6
R-0	R-0

15–8	7–0
Data 5	Data 4
R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.5.28 Update Enable Registers (DCAN IF3UPD12 to IF3UPD78)

The automatic update functionality of the IF3 register set can be configured for each message object. A message object is enabled for automatic IF3 update, if the dedicated IF3UpdEn flag is set. This means that an active NewDat flag of this message object (e.g due to reception of a CAN frame) will trigger an automatic copy of the whole message object to IF3 register set.

The update enable registers (DCAN IF3UPD12 to IF3UPD78) are shown in Table 22-41 and described in Table 22-42.

NOTE: IF3 Update enable should not be set for transmit objects.

Table 22-41. Update Enable Registers (DCAN IF3UPD12 to IF3UPD78))

Offset/Register	31/15	30/14	29/13	29/12	27/11	26/10	25/9	24/8	23/7	22/6	21/5	20/4	19/3	18/2	17/1	16/0
0x160 CAN IF3UPD12	IF3UpdEn[32:17]															
	R/W-0															
	IF3UpdEn[16:1]															
	R/W-0															
0x164 CAN IF3UPD34	IF3UpdEn[64:49]															
	R/W-0															
	IF3UpdEn[48:33]															
	R/W-0															
0x168 CAN IF3UPD56	IF3UpdEn[96:81]															
	R/W-0															
	IF3UpdEn[80:65]															
	R/W-0															
0x16C CAN IF3UPD78	IF3UpdEn[128:113]															
	R/W-0															
	IF3UpdEn[112:97]															
	R/W-0															

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 22-42. Update Enable Registers (DCAN IF3UPD12 to IF3UPD78) Field Descriptions

Bit	Field	Value	Description
31-0	IF3UpdEn[128:1]		IF3 Update Enabled (for all message objects)
		0	Automatic IF3 update is disabled for this message object.
		1	Automatic IF3 update is enabled for this message object. A message object is scheduled to be copied to IF3 register set, if NewDat flag of the message object is active.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.5.29 CAN TX I/O Control Register (DCAN TIOC)

The CAN_TX pin of the DCAN module can be used as general purpose IO pin if CAN function is not needed. The CAN TX I/O control register (DCAN TIOC) is shown in Figure 22-50 and described in Table 22-43.

NOTE: The values of the IO control registers are only writable if Init bit of the CAN control register is set.

NOTE: The OD, Func, Dir and Out bits of the CAN TX IO control register are forced to certain values when Init bit of CAN control register is reset (see bit descriptions).

Figure 22-50. CAN TX I/O Control Register (DCAN TIOC)

31	19	18	17	16
Reserved		PU	PD	OD
R-0		R/W-D	R/W-D	R/WP-0

15	3	2	1	0
Reserved	Func	Dir	Out	In
R-0	R/WP-0	R/WP-0	R/WP-0	R-U

LEGEND: R/W = Read/Write; R = Read only; WP = Protected Write (protected by Init bit), U = Undefined; D = Device dependent; -*n* = value after reset

Table 22-43. CAN TX I/O Control Register (DCAN TIOC) Field Descriptions

Bit	Field	Value	Description
31-19	Reserved	0	These bits are always read as 0. Writes have no effect.
18	PU		CAN_TX pull up/pull down select. This bit is only active when CAN_TX is configured to be an input.
		0	CAN_TX pull down is selected, when pull logic is active (PD = 0).
		1	CAN_TX pull up is selected, when pull logic is active(PD = 0).
17	PD		CAN_TX pull disable. This bit is only active when CAN_TX is configured to be an input.
		0	CAN_TX pull is active
		1	CAN_TX pull is disabled
16	OD		CAN_TX open drain enable. This bit is only active when CAN_TX is configured to be in GIO mode (TIOC.Func=0).
		0	The CAN_TX pin is configured in push/pull mode.
		1	The CAN_TX pin is configured in open drain mode.
			Forced to '0' if Init bit of CAN control register is reset.
15-4	Reserved	0	These bits are always read as 0. Writes have no effect.
3	Func		CAN_TX function. This bit changes the function of the CAN_TX pin
		0	CAN_TX pin is in GIO mode.
		1	CAN_TX pin is in functional mode (as an output to transmit CAN data).
			Forced to '1' if Init bit of CAN control register is reset.
2	Dir		CAN_TX data direction. This bit controls the direction of the CAN_TX pin when it is configured to be in GIO mode only (TIOC.Func=0)
		0	The CAN_TX pin is an input.
		1	The CAN_TX pin is an output
			Forced to '1' if Init bit of CAN control register is reset.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 22-43. CAN TX I/O Control Register (DCAN TIOC) Field Descriptions (continued)

Bit	Field	Value	Description
1	Out		CAN_TX data out write. This bit is only active when CAN_TX pin is configured to be in GIO mode (TIOC.Func = 0) and configured to be an output pin (TIOC.Dir = 1). The value of this bit indicates the value to be output to the CAN_TX pin.
		0	The CAN_TX pin is driven to logic low
		1	The CAN_TX pin is driven to logic high
			Forced to Tx output of the CAN core, if Init bit of CAN control register is reset.
0	In		CAN_TX data in
		0	The CAN_TX pin is at logic low
		1	The CAN_TX pin is at logic high
			Note: When CAN_TX pin is connected to a CAN transceiver, an external pullup resistor has to be used to ensure that the CAN bus will not be disturbed (e.g. while reset of the DCAN module).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

22.5.30 CAN RX IO Control Register (DCAN RIOC)

The CAN_RX pin of the DCAN module can be used as general purpose IO pin if CAN function is not needed. The CAN RX I/O control register (DCAN RIOC) is shown in Figure 22-51 and described in Table 22-44.

NOTE: The values of the IO control registers are writable only if Init bit of CAN control register is set.

NOTE: The OD, Func and Dir bits of the CAN RX IO control register are forced to certain values when the Init bit of CAN control register is reset (see bit descriptions).

Figure 22-51. CAN RX IO control register (DCAN RIOC)

31	19	18	17	16
Reserved		PU	PD	OD
R-0		R/W-D	R/W-D	R/WP-0

15	3	2	1	0
Reserved	Func	Dir	Out	In
R-0	R/WP-0	R/WP-0	R/WP-0	R-U

LEGEND: R/W = Read/Write; R = Read only; WP = Protected Write (protected by Init bit), U = Undefined; D = Device dependent; -*n* = value after reset

Table 22-44. CAN RX IO Control Register (DCAN RIOC) Field Descriptions

Bit	Field	Value	Description
31-19	Reserved	0	These bits are always read as 0. Writes have no effect.
18	PU		CAN_RX pull up/pull down select. This bit is only active when CAN_RX is configured to be an input.
		0	CAN_RX pull down is selected, when pull logic is active (PD = 0).
		1	CAN_T=RX pull up is selected, when pull logic is active(PD = 0).
17	PD		CAN_RX pull disable. This bit is only active when CAN_TX is configured to be an input.
		0	CAN_RX pull is active
		1	CAN_RX pull is disabled
16	OD		CAN_RX open drain enable. This bit is only active when CAN_RX is configured to be in GIO mode (TIOC.Func=0).
		0	The CAN_RX pin is configured in push/pull mode.
		1	The CAN_RX pin is configured in open drain mode.
			Forced to '0' if Init bit of CAN control register is reset.
15-4	Reserved	0	These bits are always read as 0. Writes have no effect.
3	Func		CAN_RX function. This bit changes the function of the CAN_RX pin
		0	CAN_RX pin is in GIO mode.
		1	CAN_RX pin is in functional mode (as an output to transmit CAN data).
			Forced to '1' if Init bit of CAN control register is reset.
2	Dir		CAN_RX data direction. This bit controls the direction of the CAN_RX pin when it is configured to be in GIO mode only (TIOC.Func=0)
		0	The CAN_RX pin is an input.
		1	The CAN_RX pin is an output
			Forced to '1' if Init bit of CAN control register is reset.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 22-44. CAN RX IO Control Register (DCAN RIOC) Field Descriptions (continued)

Bit	Field	Value	Description
1	Out		CAN_RX data out write. This bit is only active when CAN_RX pin is configured to be in GIO mode (TIOC.Func = 0) and configured to be an output pin (TIOC.Dir = 1). The value of this bit indicates the value to be output to the CAN_RX pin.
		0	The CAN_RX pin is driven to logic low
		1	The CAN_RX pin is driven to logic high
			Forced to Tx output of the CAN core, if Init bit of CAN control register is reset.
0	In		CAN_RX data in
		0	The CAN_RX pin is at logic low
		1	The CAN_RX pin is at logic high
			Note: When CAN_RX pin is connected to a CAN transceiver, an external pullup resistor has to be used to ensure that the CAN bus will not be disturbed (e.g. while reset of the DCAN module).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Chapter 23

SPRUH73–July 2011

Multichannel Serial Port Interface (McSPI)

This chapter describes the McSPI of the device.

Topic **Page**

23.1 Introduction **2856**
23.2 Integration **2856**
23.3 Functional Description **2859**
23.4 Use Cases **2894**
23.5 SPI Registers **2895**

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.1 Introduction

This document is intended to provide programmers with a functional presentation of the Master/Slave Multichannel Serial Port Interface (McSPI) module. It also provides a register description and a module configuration example.

McSPI is a general-purpose receive/transmit master/slave controller that can interface with up to four slave external devices or one single external master. It allows a duplex, synchronous, serial communication between a CPU and SPI compliant external devices (Slaves and Masters).

23.1.1 McSPI Features

The general features of the SPI controller are:

- Buffered receive/transmit data register per channel (1 word deep)
- Multiple SPI word access with one channel using a FIFO
- Two DMA requests per channel, one interrupt line
- Single interrupt line, for multiple interrupt source events
- Serial link interface supports:
  - Full duplex / Half duplex
  - Multi-channel master or single channel slave operations
  - Programmable 1-32 bit transmit/receive shift operations.
  - Wide selection of SPI word lengths continuous from 4 to 32 bits
- Up to four SPI channels
- SPI word Transmit / Receive slot assignment based on round robin arbitration
- SPI configuration per channel (clock definition, enable polarity and word width)
- Clock generation supports:
  - Programmable master clock generation (operating from fixed 48-MHz functional clock input)
  - Selectable clock phase and clock polarity per chip select.

23.1.2 Unsupported McSPI Features

This device supports only two chip selects per module. Module wakeup during slave mode operation is not supported, as noted in *McSPI Clock and Reset Management*.

Table 23-1. Unsupported McSPI Features

Feature	Reason
Chip selects 2 and 3	Not pinned out
Slave mode wakeup	SWAKEUP not connected
Retention during power down	Module not synthesized with retention enabled

23.2 Integration

This device includes two instantiations of McSPI: SPI0 and SPI1. The McSPI module is a general-purpose receive/transmit master/slave controller that can interface with either up to four slave external devices or one single external master. Figure 23-1 shows the example of a system with multiple external slave SPI compatible devices and Figure 23-2 shows the example of a system with an external master.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 23-1. SPI Master Application

Figure 23-2. SPI Slave Application

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.2.1 McSPI Connectivity Attributes

The general connectivity attributes for the McSPI module are shown in Table 23-2.

Table 23-2. McSPI Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_L4LS_GCLK (OCP) PD_PER_SPI_GCLK (Func)
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	1 interrupt to MPU subsystem and PRUSS (McSPI0) 1 interrupt to MPU subsystem only (McSPI1)
DMA Requests	4 DMA requests to EDMA 1 RD & 1 WR request per CS
Physical Address	L4 Peripheral slave port

23.2.2 McSPI Clock and Reset Management

The SPI module clocks can be woken up in two manners: by the SPI module itself using the SWAKEUP signal (refer to the module functional spec for detailed conditions), or directly from an external SPI master device by detecting an active low level on its chip select input pin (CS0n) using a GPIO attached to that device pin. Neither of these methods is supported on the device.

Table 23-3. McSPI Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
CLK Interface clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_l4ls_gclk From PRCM
CLKSPIREF Functional clock	48 MHz	PER_CLKOUTM2 / 4	pd_per_spi_gclk From PRCM

23.2.3 McSPI Pin List

The McSPI interface pins are summarized in Table 23-4.

Table 23-4. McSPI Pin List

Pin	Type	Description
SPI*x*_SCLK	I/O	SPI serial clock (output when master, input when slave)
SPI*x*_D0	I/O	Can be configured as either input or output (MOSI or MISO)
SPI*x*_D1	I/O	Can be configured as either input or output (MOSI or MISO)
SPI*x*_CS0n	I/O	SPI chip select 0 output when master, input when slave (active low)
SPI*x*_CS1n	O	SPI chip select 1 output when master, input when slave (active low)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.3 Functional Description

23.3.1 SPI Interface

This section lists the name and description of the SPI interface used for connection to external SPI compliant devices. McSPI has a total of seven external SPI signals shown in Table 23-5.

Table 23-5. McSPI Interface Pins

Pin	Type	Description
SPI_SCLK	I/O	SPI serial clock (output when master, input when slave)
SPI_D0	I/O	Can be configured as either input or output (MOSI or MISO)
SPI_D1	I/O	Can be configured as either input or output (MOSI or MISO)
$\overline{\text{SPI_CS0}}$	I/O	SPI chip select 0 output when master, input when slave (active low)
$\overline{\text{SPI_CS1}}$	I/O	SPI chip select 0 output when master, input when slave (active low)
$\overline{\text{SPI_CS2}}$	I/O	SPI chip select 2 output when master, input when slave (active low)
$\overline{\text{SPI_CS3}}$	I/O	SPI chip select 3 output when master, input when slave (active low)

23.3.2 SPI Transmission

This section describes the transmissions supported by McSPI. The SPI protocol is a synchronous protocol that allows a master device to initiate serial communication with a slave device. Data is exchanged between these devices. A slave select line (SPIEN) can be used to allow selection of an individual slave SPI device. Slave devices that are not selected do not interfere with SPI bus activities. Connected to multiple external devices, McSPI exchanges data with a single SPI device at a time through two main modes:

- Two data pins interface mode. (See Section 23.3.2.1)
- Single data pin interface mode (recommended for half-duplex transmission). (See Section 23.3.2.2)

The flexibility of McSPI allows exchanging data with several formats through programmable parameters described in Section 23.3.2.3.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.3.2.1 Two Data Pins Interface Mode

The two data pins interface mode, allows a full duplex SPI transmission where data is transmitted (shifted out serially) and received (shifted in serially) simultaneously on separate data lines SPIDAT [0] and SPIDAT [1]. Data leaving the master exits on transmit serial data line also known as MOSI: MasterOutSlaveIn. Data leaving the slave exits on the receive data line also known as MISO: MasterInSlaveOut.

McSPI has a unified SPI port control: SPIDAT [1:0] can be independently configured as receive or transmit lines. The user has the responsibility to program which data line to use and in which direction (receive or transmit), according to the external slave/master connection.

The serial clock (SPICLK) synchronizes shifting and sampling of the information on the two serial data lines (SPIDAT [1:0]). Each time a bit is transferred out from the Master, one bit is transferred in from Slave.

Figure 23-3 shows an example of a full duplex system with a Master device on the left and a Slave device on the right. After 8 cycles of the serial clock SPICLK, the WordA has been transferred from the master to the slave. At the same time, the WordB has been transferred from the slave to the master.

When referring to the master device, the control block transmits the clock SPICLK and the enable signal SPIEN(optional depends on MCSPI_MODULECTRL[PIN34] bit field).

Figure 23-3. SPI Full-Duplex Transmission

23.3.2.2 Single Data Pin Interface Mode

In single data pin interface mode, under software control, a single data line is used to alternatively transmit and receive data (Half duplex transmission).

McSPI has a unified SPI port control: SPIDAT [1:0] can be independently configured as receive or transmit lines. The user has the responsibility to program which data line to use and in which direction (receive or transmit), according to the external slave/master connection.

As for a full duplex transmission, the serial clock (SPICLK) synchronizes shifting and sampling of the information on the single serial data line.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.3.2.2.1 Example With a Receive-Only Slave

Figure 23-4 shows a half duplex system with a Master device on the left and a receive-only Slave device on the right. Each time a bit is transferred out from the Master, one bit is transferred in the Slave. After 8 cycles of the serial clock SPICLK, the WordA has been transferred from the master to the slave.

Figure 23-4. SPI Half-Duplex Transmission (Receive-only Slave)

23.3.2.2.2 Example With a Transmit-Only Slave

Figure 23-5 shows a half duplex system with a Master device on the left and a transmit-only Slave device on the right. Each time a bit is transferred out from the Slave, one bit is transferred in the Master. After 8 cycles of the serial clock SPICLK, the WordA has been transferred from the slave to the master.

Figure 23-5. SPI Half-Duplex Transmission (Transmit-Only Slave)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.3.2.3 Transfer Formats

This section describes the transfer formats supported by McSPI.

The flexibility of McSPI allows setting the parameters of the SPI transfer:

- SPI word length
- SPI enable generation programmable
- SPI enable assertion
- SPI enable polarity
- SPI clock frequency
- SPI clock phase
- SPI clock polarity

The consistency between SPI word length, clock phase and clock polarity of the master SPI device and the communicating slave device remains under software responsibility.

23.3.2.3.1 Programmable Word Length

McSPI supports any SPI word from 4 to 32 bits long.

The SPI word length can be changed between transmissions to allow a master device to communicate with peripheral slaves having different requirements.

23.3.2.3.2 Programmable SPI Enable Generation

McSPI is able to generate or not the SPI enable, if management of chip select is de-asserted a point to point connection is mandatory. Only a single master of slave device can be connected to the SPI bus.

23.3.2.3.3 Programmable SPI Enable (SPIEN)

The polarity of the SPIEN signals is programmable. SPIEN signals can be active high or low.

The assertion of the SPIEN signals is programmable: SPIEN signals can be manually asserted or can be automatically asserted.

Two consecutive words for two different slave devices may go along with active SPIEN signals with different polarity.

23.3.2.3.4 Programmable SPI Clock (SPICLK)

The phase and the polarity of the SPI serial clock are programmable when McSPI is a SPI master device or a SPI slave device. The baud rate of the SPI serial clock is programmable when McSPI is a SPI master.

When McSPI is operating as a slave, the serial clock SPICLK is an input from the master.

23.3.2.3.5 Bit Rate

In Master Mode, an internal reference clock CLKSPIREF is used as an input of a programmable divider to generate bit rate of the serial clock SPICLK. Granularity of this clock divider can be changed.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.3.2.3.6 Polarity and Phase

McSPI supports four sub-modes of the SPI format transfer that depend on the polarity (POL) and the phase (PHA) of the SPI serial clock (SPICLK). Table 23-6 and Figure 23-6 show a summary of the four sub-modes. Software selects one of four combinations of serial clock phase and polarity.

Two consecutive SPI words for two different slave devices may go along with active SPICLK signal with different phase and polarity.

Table 23-6. Phase and Polarity Combinations

Polarity (POL)	Phase (PHA)	SPI Mode	Comments
0	0	mode0	SPICLK active high and sampling occurs on the rising edge.
0	1	mode1	SPICLK active high and sampling occurs on the falling edge.
1	0	mode2	SPICLK active low and sampling occurs on the falling edge.
1	1	mode3	SPICLK active low and sampling occurs on the rising edge.

Figure 23-6. Phase and Polarity Combinations

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.3.2.3.7 Transfer Format With PHA = 0

This section describes the concept of a SPI transmission with the SPI mode0 and the SPI mode2.

In the transfer format with PHA = 0, SPIEN is activated a half cycle of SPICLK ahead of the first SPICLK edge.

In both master and slave modes, McSPI drives the data lines at the time of SPIEN is asserted.

Each data frame is transmitted starting with the MSB. At the extremity of both SPI data lines , the first bit of SPI word is valid a half cycle of SPICLK after the SPIEN assertion.

Therefore, the first edge of the SPICLK line is used by the master to sample the first data bit sent by the slave. On the same edge, the first data bit sent by the master is sampled by the slave.

On the next SPICLK edge, the received data bit is shifted into the shift register, and a new data bit is transmitted on the serial data line.

This process continues for a total of pulses on the SPICLK line defined by the SPI word length programmed in the master device, with data being latched on odd numbered edges and shifted on even numbered edges.

Figure 23-7 is a timing diagram of a SPI transfer for the SPI mode0 and the SPI mode2, when McSPI is master or slave, with the frequency of SPICLK equals to the frequency of CLKSPIREF. It should not be used as a replacement for SPI timing information and requirements detailed in [3].

When McSPI is in slave mode, if the SPIEN line is not de-asserted between successive transmissions then the content of the Transmitter register is not transmitted, instead the last received SPI word is transmitted.

In master mode, the SPIEN line must be negated and reasserted between each successive SPI word. This is because the slave select pin freezes the data in its shift register and does not allow it to be altered if PHA bit equals 0.

In 3-pin mode without using the SPIEN signal, the controller provides the same waveform but with SPIEN forced to low state. In slave mode SPIEN is useless

Figure 23-7. Full Duplex Single Transfer Format with PHA = 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.3.2.3.8 Transfer Format With PHA = 1

This section describes SPI full duplex transmission with the SPI mode1 and the SPI mode3.

In the transfer format with PHA = 1, SPIEN is activated a delay (t_{Lead}) ahead of the first SPICLK edge.

In both master and slave modes, McSPI drives the data lines on the first SPICLK edge.

Each data frame is transmitted starting with the MSB. At the extremity of both SPI data lines , the first bit of SPI word is valid on the next SPICLK edge, a half cycle of SPICLK later. It is the sampling edge for both the master and slave.

When the third edge occurs, the received data bit is shifted into the shift register. The next data bit of the master is provided to the serial input pin of the slave.

This process continues for a total of pulses on the SPICLK line defined by the word length programmed in the master device, with data being latched on even numbered edges and shifted on odd numbered edges.

Figure 23-8 is a timing diagram of a SPI transfer for the SPI mode1 and the SPI mode3, when McSPI is master or slave, with the frequency of SPICLK equals to the frequency of CLKSPIREF. It should not be used as a replacement for SPI timing information and requirements detailed in [3].

The SPIEN line may remain active between successive transfers. In 3-pin mode without using the SPIEN signal, the controller provides the same waveform but with SPIEN forced to low state. In slave mode SPIEN is useless.

Figure 23-8. Full Duplex Single Transfer Format With PHA = 1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.3.3 Master Mode

McSPI is in master mode when the bit MS of the register MCSPI_MODULCTRL is cleared.

In master mode McSPI supports multi-channel communication with up to 4 independent SPI communication channel contexts. McSPI initiates a data transfer on the data lines (SPIDAT [1;0]) and generates clock (SPICLK) and control signals (SPIEN) to a single SPI slave device at a time.

23.3.3.1 Dedicated Resources Per Channel

In the following sections, the letter “I” indicates the channel number that can be 0, 1, 2 or 3. Each channel has the following dedicated resources:

- Its own channel enable, programmable with the bit EN of the register MCSPI_CH(I)CTRL. Disabling the channel, outside data word transmission, remains under user responsibility.
- Its own transmitter register MCSPI_TX on top of the common shift register. If the transmitter register is empty, the status bit TXS of the register MCSPI_CH(I)STAT is set.
- Its own receiver register MCSPI_RX on top of the common shift register. If the receiver register is full, the status bit RXS of the register MCSPI_CH(I)STAT is set.
- A fixed SPI ENABLE line allocation (SPIEN[i] port for channel “I”), SPI enable management is optional.
- Its own communication configuration with the following parameters via the register (I)CONF
  - Transmit/Receive modes, programmable with the bit TRM.
  - Interface mode (Two data pins or Single data pin) and data pins assignment, both programmable with the bits IS and DPE.
  - SPI word length, programmable with the bits WL.
  - SPIEN polarity, programmable with the bit EPOL.
  - SPIEN kept active between words, programmable with the bit FORCE.
  - Turbo mode, programmable with the bit TURBO.
  - SPICLK frequency, programmable with the bit CLKD, the granularity of clock division can be changed using CLKG bit, the clock ratio is then concatenated with MCSPI_CHCTRL[EXTCLK] value.
  - SPICLK polarity, programmable with the bit POL
  - SPICLK phase, programmable with the bit PHA.
  - Start bit extension enable, programmable with the bit SBE to support LoSSI transfer specification [4].
  - Start bit polarity, programmable with the bit SBPOL
  - Use a FIFO Buffer or not( ), programmable with FFER and FFEW, depending on transfer mode TRM.
- Two DMA requests events, read and write, to synchronize read/write accesses of the DMA controller with the activity of McSPI. The DMA requests are enabled with the bits DMAR and DMAW.
- Three interrupts events

The transfers will use the latest loaded parameters of the register (I)CONF.

The configuration parameters SPIEN polarity, Turbo mode, SPICLK phase and SPICLK polarity can be loaded in the (I)CONF register only when the channel is disabled. The user has the responsibility to change the other parameters of the (I)CONF register when no transfer occurs on the SPI interface.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.3.3.2 Interrupt Events in Master Mode

In master mode, the interrupt events related to the transmitter register state are TX_empty and TX_underflow. The interrupt event related to the receiver register state is RX_full.

23.3.3.2.1 TX_empty

The event TX_empty is activated when a channel is enabled and its transmitter register becomes empty (transient event). Enabling channel automatically raises this event, except for the Master receive only mode. (See Section 23.3.3.5). When FIFO buffer is enabled (MCSPI_CH(I)CONF[FFEW] set to 1), the TX_empty is asserted as soon as there is enough space in buffer to write a number of byte defined by MCSPI_XFERLEVEL[AEL].

Transmitter register must be loaded to remove the source of the interrupt and the TX_empty interrupt status bit must be cleared for interrupt line de-assertion (if event enabled as interrupt source) . (See Section 23.3.5).

When FIFO is enabled, no new TX_empty event will be asserted as soon as CPU has not performed the number of write into transmitter register defined by MCSPI_XFERLEVEL[AEL]. It is the responsibility of CPU to perform the right number of writes.

23.3.3.2.2 TX_underflow

The event TX_underflow is activated when the channel is enabled and if the transmitter register or FIFO is empty (not updated with new data) at the time of shift register assignment.

The TX_underflow is a harmless warning in master mode.

To avoid having TX_underflow event at the beginning of a transmission, the event TX_underflow is not activated when no data has been loaded into the transmitter register since channel has been enabled.

To avoid having TX_underflow event the Transmitter register must be loaded seldom.

TX_underflow interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).

Note: Be careful, when more than one channel have a FIFO enable bit field FFER or FFEW set, the module force no use of FIFO, software must take care that only one enabled channel is configured to use the FIFO buffer.

23.3.3.2.3 RX_ full

The event RX_full is activated when channel is enabled and receiver register becomes filled (transient event). When FIFO buffer is enabled (MCSPI_CH(I)CONF[FFER] set to 1), the RX_full is asserted as soon as there is a number of bytes holds in buffer to read defined by MCSPI_XFERLEVEL[AFL].

Receiver register must be read to remove source of interrupt and RX_full interrupt status bit must be cleared for interrupt line de-assertion (if event enabled as interrupt source).

When FIFO is enabled, no new RX_full event will be asserted as soon as CPU has not performed the number of read into receive register defined by MCSPI_XFERLEVEL[AFL]. It is the responsibility of CPU to perform the right number of reads.

23.3.3.2.4 End of Word Count

The event end of word count (EOW) is activated when channel is enabled and configured to use the built-in FIFO. This interrupt is raised when the controller had performed the number of transfer defined in MCSPI_XFERLEVEL[WCNT] register. If the value was programmed to 0000h, the counter is not enabled and this interrupt is not generated.

The End of Word count interrupt also indicates that the SPI transfer is halt on channel using the FIFO buffer as soon as MCSPI_XFERLEVEL[WCNT] is not reloaded and channel re-enabled.

End of Word interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.3.3.3 Master Transmit and Receive Mode

This mode is programmable per channel (bits TRM of the register (I)CONF).

The channel access to the shift registers, for transmission/reception, is based on its transmitter and receiver register state and round robin arbitration.

The channel that meets the rules below is included in the round robin list of already active channels scheduled for transmission and/or reception. The arbiter skips the channel that does not meet the rules and search for the next following enabled channel, in rotation.

Rule 1: Only enabled channels (bit EN of the register MCSPI_CH(I)CTRL), can be scheduled for transmission and/or reception.

Rule 2: An enabled channel can be scheduled if its transmitter register is not empty (bit TXS of the register MCSPI_CH(I)STAT) or its FIFO is not empty in case of buffer use for the corresponding channel (bit FFE of the register MCSPI_CH(I)STAT), that is updated with new data, at the time of shift register assignment. If the transmitter register or FIFO is empty, at the time of shift register assignment, the event TX_underflow is activated and the next enabled channel with new data to transmit is scheduled. (See also transmit only mode).

Rule 3: An enabled channel can be scheduled if its receive register is not full (bit RXS of the register MCSPI_CH(I)STAT)) or its FIFO is not full in case of buffer use for the corresponding channel (bit FFF of the register MCSPI_CH(I)STAT) at the time of shift register assignment. (See also receive only mode). Therefore the receiver register of FIFO cannot be overwritten. The RX_overflow bit, in the MCSPI_IRQSTATUS register is never set in this mode.

On completion of SPI word transfer (bit EOT of the register MCSPI_CH(I)STAT is set) the updated transmitter register of the next scheduled channel is loaded into the shift register. This bit is meaningless when using the Buffer for this channel. The serialization (transmit and receive) starts according to the channel communication configuration. On serialization completion the received data is transferred to the channel receive register.

The built-in FIFO is available in this mode and can be configured in one data direction Transmit or Receive, then the FIFO is seen as a unique FFNBYTE bytes buffer, or it can also be configured in both data direction Transmit and Receive, then the FIFO is split into two separate FFNBYTE/2 bytes buffer with their own address space management, in this last case the definition of AEL and AFL levels is based on FFNBYTE/2 bytes and is under CPU responsibility.

23.3.3.4 Master Transmit-Only Mode

This mode avoids the CPU to read the receiver register (minimizing data movement) when only transmission is meaningful.

The master transmit only mode is programmable per channel (bits TRM of the register (I)CONF).

In master transmit only mode, transmission starts after data is loaded into the transmitter register.

Rule 1 and **Rule 2**, defined above, are applicable in this mode.

Rule 3, defined above, is not applicable: In master transmit only mode, the receiver register or FIFO state “full” does not prevent transmission, and the receiver register is always overwritten with the new SPI word. This event in the receiver register is not significant when only transmission is meaningful. So, the RX_overflow bit, in the MCSPI_IRQSTATUS register is never set in this mode.

The McSPI module automatically disables the RX_full interrupt status. The corresponding interrupt request and DMA Read request are not generated in master transmit only mode.

The status of the serialization completion is given by the bit EOT of the register MCSPI_CH(I)STAT. This bit is meaningless when using the Buffer for this channel.

The built-in FIFO is available in this mode and can be configured with FFEW bit field in the MCSPI_CH(I)CONF register, then the FIFO is seen as a unique FFNBYTE bytes buffer.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.3.3.5 Master Receive-Only Mode

This mode avoids the CPU to refill the transmitter register (minimizing data movement) when only reception is meaningful.

The master receive mode is programmable per channel (bits TRM of the register (I)CONF).

The master receive only mode enables channel scheduling only on empty state of the receiver register.

Rule 1 and **Rule 3**, defined above, are applicable in this mode.

Rule 2, defined above, is not applicable: In master receive only mode, after the first loading of the transmitter register of the enabled channel, the transmitter register state is maintained as full. The content of the transmitter register is always loaded into the shift register, at the time of shift register assignment. So, after the first loading of the transmitter register, the bits TX_empty and TX_underflow, in the MCSPI_IRQSTATUS register are never set in this mode.

The status of the serialization completion is given by the bit EOT of the register MCSPI_CH(I)STAT. The bit RX_full in the MCSPI_IRQSTATUS register is set when a received data is loaded from the shift register to the receiver register. This bit is meaningless when using the Buffer for this channel.

The built-in FIFO is available in this mode and can be configured with FFER bit field in the MCSPI_CH(I)CONF register, then the FIFO is seen as a unique FFNBYTE bytes buffer.

23.3.3.6 Single-Channel Master Mode

When the SPI is configured as a master device with a single enabled channel, the assertion of the SPIM_CSX signal can be controlled in two different ways:

- In 3 pin mode : MCSPI_MODULCTRL[1] PIN34 and MCSPI_MODULCTRL[0] SINGLE bit are set to 1, the controller transmit SPI word as soon as transmit register or FIFO is not empty.
- In 4 pin mode : MCSPI_MODULCTRL[1] PIN34 bit is cleared to 0 and MCSPI_MODULCTRL[0] SINGLE bit is set to 1, SPIEN assertion/deassertion controlled by Software. (See Section 23.3.3.6.1) using the MCSPI_CHxCONF[20] FORCE bit.

23.3.3.6.1 Programming Tips When Switching to Another Channel

When a single channel is enabled and data transfer is ongoing:

- Wait for completion of the SPI word transfer (bit EOT of the register MCSPI_CH(I)STAT is set) before disabling the current channel and enabling a different channel.
- Disable the current channel first, and then enable the other channel.

23.3.3.6.2 Keep SPIEN Active Mode (Force SPIEN)

Continuous transfers are manually allowed by keeping the SPIEN signal active for successive SPI words transfer. Several sequences (configuration/enable/disable of the channel) can be run without deactivating the SPIEN line. This mode is supported by all channels and any master sequence can be used (transmit-receive, transmit-only, receive-only).

Keeping the SPIEN active mode is supported when:

- A single channel is used (bit MCSPI_MODULCTRL[Single] is set to 1).
- Transfer parameters of the transfer are loaded in the configuration register (MCSPI_CH(I)CONF) in the appropriate channel.

  The state of the SPIEN signal is programmable.
  - Writing 1 into the bit FORCE of the register MCSPI_CH(I)CONF drives high the SPIEN line when MCSPI_CHCONF(I)[EPOL] is set to zero, and drives it low when MCSPI_CHCONF(I)[EPOL] is set.
  - Writing 0 into the bit FORCE of the register MCSPI_CH(I)CONF drives low the SPIEN line when MCSPI_CHCONF(I)[EPOL] is set to zero, and drives it high when MCSPI_CHCONF(I)[EPOL] is set.
- A single channel is enabled (MCSPI_CH(I)CTRL[En] set to 1) . The first enabled channel activates the SPIEN line.

Copyright © 2011, Texas Instruments Incorporated

Once the channel is enabled, the SPIEN signal is activated with the programmed polarity.

As in multi-channel master mode, the start of the transfer depends on the status of the transmitter register, the status of the receiver register and the mode defined by the bits TRM in the configuration register (transmit only, receive only or transmit and receive) of the enabled channel.

The status of the serialization completion of each SPI word is given by the bit EOT of the register MCSPI_CH(I)STAT. The bit RX_full in the MCSPI_IRQSTATUS register is set when a received data is loaded from the shift register to the receiver register.

A change in the configuration parameters is propagated directly on the SPI interface. If the SPIEN signal is activated the user must insure that the configuration is changed only between SPI words, in order to avoid corrupting the current transfer.

NOTE: The SPIEN polarity, the SPICLK phase and SPICLK polarity must not be modified when the SPIEN signal is activated. The Transmit/Receive mode, programmable with the bit TRM can be modified only when the channel is disabled. The channel can be disabled and enabled while the SPIEN signal is activated.

The delay between SPI words that requires the connected SPI slave device to switch from one configuration (Transmit only for instance) to another (receive only for instance) must be handled under software responsibility.

At the end of the last SPI word, the channel must be deactivated (MCSPI_CH(I)CTRL[En] is cleared to 0) and the SPIEN can be forced to its inactive state (MCSPI_CH(I)CONF[Force]).

Figure 23-9 and Figure 23-10 show successive transfers with SPIEN kept active low with a different configuration for each SPI word in respectively single data pin interface mode and two data pins interface mode. The arrows indicate when the channel is disabled before a change in the configuration parameters and enabled again.

Figure 23-9. Continuous Transfers With SPIEN Maintained Active (Single-Data-Pin Interface Mode)

Figure 23-10. Continuous Transfers With SPIEN Maintained Active (Dual-Data-Pin Interface Mode)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

NOTE: The turbo mode is also supported for the Keep SPIEN active mode when the following conditions are met:

- A single channel will be explicitly used (bit MCSPI_MODULCTRL[Single] is set to 1).
- The turbo mode is enabled in the configuration of the channel (bit Turbo of the register (i)CONF).

23.3.3.6.3 Turbo Mode

The purpose of the Turbo mode is to improve the throughput of the SPI interface when a single channel is enabled, by allowing transfers until the shift register and the receiver register are full.

This mode is programmable per channel (bit Turbo of the register (I)CONF). When several channels are enabled, the bit Turbo of the registers MCSPI_CH(I)CONF has no effect, and the channel access to the shift registers remains as described in Section 23.3.3.3.

In Turbo mode, **Rule 1** and **Rule 2** defined in Section 23.3.3.3 are applicable but Rule 3 is not applicable. An enabled channel can be scheduled if its receive register is full (bit RXS of the register MCSPI_CH(I)STAT) at the time of shift register assignment until the shift register is full.

In Turbo mode, **Rule 1** and **Rule 2** defined in Section 23.3.3.3 are applicable but Rule 3 is not applicable. An enabled channel can be scheduled if its receive register is full (bit RXS of the register MCSPI_CH(I)STAT) at the time of shift register assignment until the shift register is full.

The receiver register cannot be overwritten in turbo mode. In consequence the RX_overflow bit, in MCSPI_IRQSTATUS register is never set in this mode.

23.3.3.7 Start Bit Mode

The purpose of the start bit mode is to add an extended bit before the SPI word transmission specified by word length WL. This feature is only available in master mode. This mode is compliant with write command/data format as specified in LoSSI protocol.

This mode is programmable per channel (bit Start bit enable SBE of the register MCSPI_CH(I)CONF).

The polarity of the extended bit is programmable per channel and it indicates whether the next SPI word must be handled as a command when SBPOL is cleared to 0 or as a data or a parameter when SBPOL is set to 1 (refer to LoSSI specification [4]). Moreover start bit polarity SBPOL can be changed dynamically during start bit mode transfer without disabling the channel for reconfiguration, in this case you have the responsibility to configure the SBPOL bit before writing the SPI word to be transmitted in TX register.

The start bit mode could be used at the same time as turbo mode and/or manual chip select mode. In this case only one channel could be used, no round-robin arbitration is possible.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 23-11. Extended SPI Transfer With Start Bit PHA = 1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.3.3.8 Chip-Select Timing Control

The chip select timing control is only available in master mode with automatic chip select generation (FORCE bit field is cleared to 0), to add a programmable delay between chip select assertion and first clock edge or chip select removal and last clock edge. The option is available only in 4 pin mode MCSPI_MODULCTRL[1] PIN34 is cleared to 0.

This mode is programmable per channel (bit TCS of the register MCSPI_CH(I)CONF). Figure 23-12 shows the chip-select SPIEN timing control.

Figure 23-12. Chip-Select SPIEN Timing Controls

NOTE: Because of the design implementation for transfers using a clock divider ratio set to 1 (clock bypassed), a half cycle must be added to the value between chip-select assertion and the first clock edge with PHA = 1 or between chip-select removal and the last clock edge with PHA = 0.

With an odd clock divider ratio which occurs when granularity is one clock cycle, that means that MCSPI_CH(I)CONF[CLKG] is set to 1 and MCSPI_CH(I)CONF[CLKD] has an even value, the clock duty cycle is not 50%, then one of the high level or low level duration is selected to be added to TCS delay.

Table 23-7 summarizes all delays between chip select and first (setup) or last (hold) clock edge.

In 3-pin mode this option is useless, the chip select SPIEN is forced to low state.

Table 23-7. Chip Select ↔ Clock Edge Delay Depending on Configuration

Clock Ratio F_{ratio}	Clock Phase PHA	Chip Select ↔ Clock Edge Delay	
		Setup	Hold
1	0	T_ref × (TCS + ½)	T_ref × (TCS + 1)
	1	T_ref × (TCS + 1)	T_ref × (TCS + ½)
Even ≥ 2	x	T_ref × F_{ratio} × (TCS + ½)	T_ref × F_{ratio} × (TCS + ½)
Odd ≥ 3 (only with MCSPI_CH(I)CONF[CLKG] set to 1	0	T_ref × [{F_{ratio} × TCS) + (F_{ratio} + ½)]	T_ref × [{F_{ratio} × TCS) + (F_{ratio} + ½)]
	1	T_ref × [{F_{ratio} × TCS) + (F_{ratio} - ½)]	T_ref × [{F_{ratio} × TCS) + (F_{ratio} - ½)]

T_ref = CLKSPIREF period in ns. F_{ratio} = SPI clock division ratio

The clock divider ratio depends on divider granularity MCSPI_CH(I)CONF[CLKG]:

- MCSPI_CH(I)CONF[CLKG] = 0 : granularity is power of two.
  $F_{ratio} = 2^{MCSPI_CH(I)CONF[CLKD]}$
- MCSPI_CH(I)CONF[CLKG] = 0 : granularity is power of two.

Copyright © 2011, Texas Instruments Incorporated

F_{ratio} = MCSPI_CH(I)CNTRL[EXTCLK].MCSPI_CH(I)CONF[CLKD] + 1

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.3.3.9 Clock Ratio Granularity

By default the clock division ratio is defined by the register MCSPI_CH(I)CONF[CLKD] with power of two granularity leading to a clock division in range 1 to 32768, in this case the duty cycle is always 50%. With bit MCSPI_CH(I)CONF[CLKG] the clock division granularity can be changed to one clock cycle, in that case the register MCSPI_CH(I)CTRL[EXTCLK] is concatenated with MCSPI_CH(I)CONF[CLKD] to give a 12-bit width division ratio in range 1 to 4096.

When granularity is one clock cycle (MCSPI_CH(I)CONF[CLKG] set to 1), for odd value of clock ratio the clock duty cycle is kept to 50-50 using falling edge of clock reference CLKSPIREF.

Table 23-8. CLKSPIO High/Low Time Computation

Clock Ratio F_{ratio}	CLKSPIO High Time	CLKSPIO Low Time
1	T_{high}_ref	T_{low}_ref
Even ≥ 2	t_ref × (F_{ratio}/2)	t_ref × (F_{ratio}/2)
Odd ≥ 3	t_ref × (F_{ratio}/2)	t_ref × (F_{ratio}/2)

T_ref = CLKSPIREF period in ns. T_{high}_ref = CLKSPIREF high Time period in ns. T_{low}_ref = CLKSPIREF low Time period in ns. F_{ratio} = SPI clock division ratio

F_{ratio} = MCSPI_CH(I)CTRL[EXTCLK].MCSPI_CH(I)CONF[CLKD] + 1

For odd ratio value the duty cycle is calculated as below:

Duty_cycle = ½

Granularity examples: With a clock source frequency of 48 MHz:

Table 23-9. Clock Granularity Examples

MCSPI_CH (I)CTRL	MCSPI_CH (I)CONF	MCSPI_CH (I)CONF		MCSPI_CH (I)CONF	MCSPI_CH (I)CONF	Thigh (ns)	Tlow (ns)	Tperiod (ns)	Duty Cycle	Fout (MHz)
EXTCLK	CLKD	CLKG	F_{ratio}	PHA	POL					
X	0	0	1	X	X	10.4	10.4	20.8	50-50	48
X	1	0	2	X	X	20.8	20.8	41.6	50-50	24
X	2	0	4	X	X	41.6	41.6	83.2	50-50	12
X	3	0	8	X	X	83.2	83.2	166.4	50-50	6
0	0	1	1	X	X	10.4	10.4	20.8	50-50	48
0	1	1	2	X	X	20.8	20.8	41.6	50-50	24
0	2	1	3	1	0	31,2	31,2	62.4	50-50	16
0	2	1	3	1	1	31,2	31,2	62.4	50-50	16
0	3	1	4	X	X	41.6	41.6	83.2	50-50	12
5	0	1	81	1	0	842,4	842,4	1684.8	50-50	0.592
5	7	1	88	X	X	915.2	915.2	1830.4	50-50	0.545

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.3.3.10 FIFO Buffer Management (Optional USEFIFO = 1)

The McSPI controller has a built-in FFNBYTE bytes buffer in order to unload DMA or interrupt handler and improve data throughput. The instantiation of this Buffer is optional it depends on a generic parameter USEFIFO, the FIFO is instantiated when it is set to 1. Allowed FIFO depth up to 64 bytes is supported and is defined by generic parameter FFNBYTE. When the FIFO is not instantiated writes in registers MCSPI_XFERLEVEL, MCSPI_CH(I)CONF[FFER] and MCSPI_CH(I)CONF[FFEW] have no functional effect, nevertheless read back is allowed to check written value.

This buffer can be used by only one channel at once and is selected by setting MCSPI_CH(I)CONF[FFER] or MCSPI_CH(I)CONF[FFEW] to 1.

If several channel are selected and several FIFO enable bitfields set to 1, the controller forces buffer not to be used, it is the responsibility of the driver to set only one FIFO enable bitfield.

The buffer can be used in the modes defined below:

- Master or Slave mode.
- Transmit only, Receive only or Transmit/Receive mode.
- Single channel or turbo mode, or in normal round robin mode. In round robin mode the buffer is used by only one channel.
- Every word length MCSPI_CH(I)CONF[WL] are supported.

Two levels AEL and AFL located in MCSPI_XFERLEVEL register rule the buffer management. The granularity of these levels is one byte, then it is not aligned with SPI word length. It is the responsibility of the driver to set these values as a multiple of SPI word length defined in MCSPI_CH(I)CONF[WL]. The number of byte written in the FIFO depends on word length (see Table 23-10).

Table 23-10. FIFO Writes, Word Length Relationship

	SPI Word Length WL		
	3 ≤ WL ≤ 7	**8 ≤ WL ≤ 15**	**16 ≤ WL ≤ 31**
Number of byte written in the FIFO	1 byte	2 bytes	4 byte

23.3.3.10.1 Splitted FIFO

The FIFO can be splitted into two part when module is configured in transmit/receive mode MCSPI_CH(I)CONF[TRM] is cleared to 0 and MCSPI_CH(I)CONF[FFER] and MCSPI_CH(I)CONF[FFEW] asserted. Then system can access a FFNBYTE/2 byte depth FIFO per direction.

The FIFO buffer pointers are reset when the corresponding channel is enabled or FIFO configuration changes.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 23-13. Transmit/Receive Mode With No FIFO Used

Configuration:
MCSPI_CH(i)CONF[TRM]=0x0 Transmit/receive mode
MCSPI_CH(i)CONF[FFRE]=0x0 FIFO disabled on receive path
MCSPI_CH(i)CONF[FFWE]=0x0 FIFO disabled on transmit path

Figure 23-14. Transmit/Receive Mode With Only Receive FIFO Enabled

Configuration:
MCSPI_CH(i)CONF[TRM]=0x0 Transmit/receive mode
MCSPI_CH(i)CONF[FFRE]=0x1 FIFO enabled on receive path
MCSPI_CH(i)CONF[FFWE]=0x0 FIFO disabled on transmit path

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 23-15. Transmit/Receive Mode With Only Transmit FIFO Used

Configuration:
MCSPI_CH(i)CONF[TRM]=0x0 Transmit/receive mode
MCSPI_CH(i)CONF[FFRE]=0x0 FIFO disabled on receive path
MCSPI_CH(i)CONF[FFWE]=0x1 FIFO enabled on transmit path

Figure 23-16. Transmit/Receive Mode With Both FIFO Direction Used

Configuration:
MCSPI_CH(i)CONF[TRM]=0x0 Transmit/receive mode
MCSPI_CH(i)CONF[FFRE]=0x1 FIFO enabled on receive path
MCSPI_CH(i)CONF[FFWE]=0x0 FIFO disabled on transmit path

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Figure 23-17. Transmit-Only Mode With FIFO Used

Configuration:
MCSPI_CH(i)CONF[TRM]=0x2 Transmit only mode
MCSPI_CH(i)CONF[FFRE]=0x1 FIFO enabled on transmit path
MCSPI_CH(i)CONF[FFWE] not applicable

Figure 23-18. Receive-Only Mode With FIFO Used

Configuration:
MCSPI_CH(i)CONF[TRM]=012 Receive only mode
MCSPI_CH(i)CONF[FFRE]=0x1 FIFO enabled on receive path
MCSPI_CH(i)CONF[FFWE] not applicable

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.3.3.10.2 Buffer Almost Full

The bitfield MCSPI_XFERLEVEL[AFL] is needed when the buffer is used to receive SPI word from a slave (MCSPI_CH(I)CONF[FFER] must be set to 1) and register width depends on the generic parameter FFNBYTE value. It defines the almost full buffer status.

When FIFO pointer reaches this level an interrupt or a DMA request is sent to the CPU to enable system to read AFL+1 bytes from receive register. Be careful AFL+1 must correspond to a multiple value of MCSPI_CH(I)CONF[WL].

When DMA is used, the request is de-asserted after the first receive register read.

No new request will be asserted again as long as system has not performed the right number of read accesses.

Figure 23-19. Buffer Almost Full Level (AFL)

* non-DMA mode only. In DMA mode, the DMA RX request is asserted to its active level under identical conditions.

.

NOTE: SPI_IRQSTATUS register bits are not available in DMA mode. In DMA mode, the SPIm_DMA_RX*n* request is asserted on the same conditions than the SPI_IRQSTATUS RX*n*_FULL flag.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.3.3.10.3 Buffer Almost Empty

The bitfield MCSPI_XFERLEVEL[AEL] is needed when the buffer is used to transmit SPI word to a slave (MCSPI_CH(I)CONF[FFEW]must be set to 1) and register width depends on the generic parameter FFNBYTE value. It defines the almost empty buffer status.

When FIFO pointer has not reached this level an interrupt or a DMA request is sent to the CPU to enable system to write AEL+1 bytes to transmit register. Be careful AEL+1 must correspond to a multiple value of MCSPI_CH(I)CONF[WL].

When DMA is used, the request is de-asserted after the first transmit register write.

No new request will be asserted again as long as system has not performed the right number of write accesses.

Figure 23-20. Buffer Almost Empty Level (AEL)

* non-DMA mode only. In DMA mode, the DMA TX request is asserted to its active level under identical conditions.

.

23.3.3.10.4 End of Transfer Management

When the FIFO buffer is enabled for a channel, the user shall previously configure the MCSPI_XFERLEVEL register the AEL and AFL levels and especially the WCNT bit field to define the number of Spi word to be transferred using the FIFO before enabling the channel.

This counter allows the controller to stop the transfer correctly after a defined number of Spi word transfer. If WNCT is cleared to 0, the counter is not used and the user must stop the transfer manually by disabling the channel, in this case the user doesn't know how many SPI transfers have been done. For receive transfer, software shall poll the corresponding FFE bit field and read the Receive register to empty the FIFO buffer.

When End Of Word count interrupt is generated, the user can disable the channel and poll on MCSPI_CH(I)STAT[FFE] register to know if SPI word is still there in FIFO buffer and read last words.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.3.3.10.5 Multiple SPI Word Access

The CPU has the ability to perform multiple SPI word access to the receive or transmit registers within a single 32-bit OCP access by setting the bit field MCSPI_MODULCTRL[MOA] to '1' under specific conditions:

- The channel selected has the FIFO enable.
- Only FIFO sense enabled support the kind of access.
- The bit field MCSPI_MODULCTRL[MOA] is set to 1
- Only 32-bit OCP access and data width can be performed to receive or transmit registers, for other kind of access the CPU must de-assert MCSPI_MODULCTRL[MOA] bit fields.
- The Level MCSPI_XFERLEVEL[AEL] and MCSPI_XFERLEVEL[AFL] must be 32-bit aligned , it means that AEL[0] = AEL[1] = 1 or AFL[0] = AFL[1] = 1.
- If MCSPI_XFERLEVEL[WCNT] is used it must be configured according to SPI word length.
- The word length of SPI words allows to perform multiple SPI access, that means that MCSPI_CH(I)CONF[WL] < 16

Number of SPI word access depending on SPI word length:

- 3 ≤ WL ≤ 7, SPI word length smaller or equal to byte length, four SPI words accessed per 32-bit OCP read/write. If word count is used (MCSPI_XFERLEVEL[WCNT]), set the bit field to WCNT[0]=WCNT[1]=0
- 8 ≤ WL ≤ 15, SPI word length greater than byte or equal to 16-bit length, two SPI words accessed per 32-bit OCP read/write. If word count is used (MCSPI_XFERLEVEL[WCNT]), set the bit field to WCNT[0]= 0.
- 16 ≤ WL multiple SPI word access not applicable.

23.3.3.11 First SPI Word Delayed

The McSpi controller has the ability to delay the first SPI word transfer to give time for system to complete some parallel processes or fill the FIFO in order to improve transfer bandwidth. This delay is applied only on first SPI word after SPI channel enabled and first write in Transmit register. It is based on output clock frequency.

This option is meaningful in master mode and single channel mode MCSPI_MODULECTRL[SINGLE] asserted.

Figure 23-21. Master Single Channel Initial Delay

Few delay values are available: No delay, 4/8/16/32 Spi cycles.

Its accuracy is half cycle in clock bypass mode and depends on clock polarity and phase.

Copyright © 2011, Texas Instruments Incorporated

23.3.3.12 3- or 4-Pin Mode

External SPI bus interface can be configured to use a restricted set of pin using the bit field MCSPI_MODULCTRL[1] PIN34 and depending on targeted application:

- If MCSPI_MODULECTRL[34] is cleared to 0 (default value) the controller is in 4-pin mode using the SPI pins CLKSPI, SOMI, SIMO and chip enable CS.
- If MCSPI_MODULECTRL[34] is set to 1 the controller is in 3-pin mode using the SPI pins CLKSPI, SOMI and SIMO.

In this mode it is mandatory to put the controller in single channel master mode (MCSPI_MODULECTRL[SINGLE] asserted) and to connect only one SPI device on the bus.

Figure 23-22. 3-Pin Mode System Overview

In 3-pin mode all options related to chip select management are useless:

- MCSPI_CHxCONF[EPOL]
- MCSPI_CHxCONF[TCS0]
- MCSPI_CHxCONF[FORCE]

The chip select pin SPIEN is forced to ‘0’ in this mode.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.3.4 Slave Mode

McSPI is in slave mode when the bit MS of the register MCSPI_MODULCTRL is set.

In slave mode, McSPI can be connected to up to 4 external SPI master devices. McSPI handles transactions with a single SPI master device at a time.

In slave mode, McSPI initiates data transfer on the data lines (SPIDAT[1;0]) when it receives an SPI clock (SPICLK) from the external SPI master device.

The controller is able to work with or without a chip select SPIEN depending on MCSPI_MODULCTRL[1] PIN34 bit setting. It also support transfers without dead cycle between two successive words.

23.3.4.1 Dedicated Resources

In slave mode, enabling a channel that is not channel 0 has no effect. Only channel 0 can be enabled. The channel 0, in slave mode has the following resources:

- Its own channel enable, programmable with the bit EN of the register MCSPI_CH0CTRL. This channel should be enabled before transmission and reception. Disabling the channel, outside data word transmission, remains under user responsibility.
- Any of the 4 ports SPIEN[3:0] can be used as a slave SPI device enable. This is programmable with the bits SPIENSLV of the register MCSPI_CH0CONF.
- Its own transmitter register MCSPI_TX on top of the common shift register. If the transmitter register is empty, the status bit TXS of the register MCSPI_CH0STAT is set. When McSPI is selected by an external master (active signal on the SPIEN port assigned to channel 0), the transmitter register content of channel0 is always loaded in shift register whether it has been updated or not. The transmitter register should be loaded before McSPI is selected by a master.
- Its own receiver register MCSPI_RX on top of the common shift register. If the receiver register is full, the status bit RXS of the register MCSPI_CH0STAT is set.

NOTE: The transmitter register and receiver registers of the other channels are not used. Read from or Write in the registers of a channel that is not channel 0 has no effect.

- Its own communication configuration with the following parameters via the register MCSPI_CH0CONF:
  - Transmit/Receive modes, programmable with the bit TRM.
  - Interface mode (Two data pins or Single data pin) and data pins assignment, both programmable with the bits IS and DPE.
  - SPI word length, programmable with the bits WL.
  - SPIEN polarity, programmable with the bit EPOL.
  - SPICLK polarity, programmable with the bit POL.
  - SPICLK phase, programmable with the bit PHA.
  - Use a FIFO buffer or not, programmable with FFER and FFEW, depending on transfer mode TRM.

The SPICLK frequency of a transfer is controlled by the external SPI master connected to McSPI. The bits CLKD0 of the register 0CONF are not used in slave mode.

NOTE: The configuration of the channel can be loaded in the 0CONF register only when the channel is disabled.

- Two DMA requests events, read and write, to synchronize read/write accesses of the DMA controller with the activity of McSPI. The DMA requests are enabled with the bits DMAR and DMAW of the register 0CONF.
- Four interrupts events.

Figure 23-23 shows an example of four slaves wired on a single master device.

Copyright © 2011, Texas Instruments Incorporated

Figure 23-23. Example of SPI Slave with One Master and Multiple Slave Devices on Channel 0

23.3.4.2 Interrupt Events in Slave Mode

The interrupt events related to the transmitter register state are TX_empty and TX_underflow. The interrupt events related to the receiver register state are RX_full and RX_overflow.

23.3.4.2.1 TX_EMPTY

The event TX_empty is activated when the channel is enabled and its transmitter register becomes empty. Enabling channel automatically raises this event. When FIFO buffer is enabled (MCSPI_CH(I)CONF[FFEW] set to 1), the TX_empty is asserted as soon as there is enough space in buffer to write a number of byte defined by MCSPI_XFERLEVEL[AEL].

Transmitter register must be load to remove source of interrupt and TX_empty interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).

When FIFO is enabled, no new TX_empty event will be asserted as soon as Local Host has not performed the number of write into transmitter register defined by MCSPI_XFERLEVEL[AEL]. It is the responsibility of Local Host to perform the right number of writes.

Copyright © 2011, Texas Instruments Incorporated

23.3.4.2.2 TX_UNDERFLOW

The event TX_underflow is activated when channel is enabled and if the transmitter register or FIFO (if use of buffer is enabled) is empty (not updated with new data) when an external master device starts a data transfer with McSPI (transmit and receive).

When FIFO is enabled the data emitted while underflow event is raised is not the last data written in the FIFO.

The TX_underflow indicates an error (data loss) in slave mode.

To avoid having TX_underflow event at the beginning of a transmission, the event TX_underflow is not activated when no data has been loaded into the transmitter register since channel has been enabled.

TX_underflow interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).

23.3.4.2.3 RX_FULL

The event RX_full is activated when channel is enabled and receiver becomes filled (transient event). When FIFO buffer is enabled (MCSPI_CH(I)CONF[FFER] set to 1), the RX_full is asserted as soon as there is a number of bytes holds in buffer to read defined by MCSPI_XFERLEVEL[AFL].

Receiver register must be read to remove source of interrupt and RX_full interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).

When FIFO is enabled, no new RX_full event will be asserted as soon as Local Host has not performed the number of read into receive register defined by MCSPI_XFERLEVEL[AFL]. It is the responsibility of Local Host to perform the right number of reads.

23.3.4.2.4 RX_OVERFLOW

The RX0_OVERFLOW event is activated in slave mode in either transmit-and-receive or receive-only mode, when a channel is enabled and the SPI_RX*n* register or FIFO is full when a new SPI word is received. The SPI_RX*n* register is always overwritten with the new SPI word. If the FIFO is enabled data within the FIFO are overwritten, it must be considered as corrupted. The RX0_OVERFLOW event should not appear in slave mode using the FIFO.

The RX0_OVERFLOW indicates an error (data loss) in slave mode.

The SPI_IRQSTATUS[3] RX0_OVERFLOW interrupt status bit must be cleared for interrupt line deassertion (if the event is enabled as the interrupt source).

23.3.4.2.5 End of Word Count

The event EOW (End Of Word count) is activated when channel is enabled and configured to use the built-in FIFO. This interrupt is raised when the controller had performed the number of transfer defined in MCSPI_XFERLEVEL[WCNT] register. If the value was programmed to 0000h, the counter is not enabled and this interrupt is not generated.

The End of Word count interrupt also indicates that the SPI transfer is halt on channel using the FIFO buffer as soon as MCSPI_XFERLEVEL[WCNT] is not reloaded and channel re-enabled.

End of Word interrupt status bit must be cleared for interrupt line de-assertion (if event enable as interrupt source).

23.3.4.3 Slave Transmit-and-Receive Mode

The slave transmit and receive mode is programmable (bits TRM cleared to 0 in the register MCSPI_CH(I)CONF).

After the channel is enabled, transmission and reception proceed with interrupt and DMA request events.

In slave transmit and receive mode, transmitter register should be loaded before McSPI is selected by an external SPI master device.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Transmitter register or FIFO (if use of buffer enabled) content is always loaded in shift register whether it has been updated or not. The event TX_underflow is activated accordingly, and does not prevent transmission.

On completion of SPI word transfer (bit EOT of the register MCSPI_CH(I)STAT is set) the received data is transferred to the channel receive register. This bit is meaningless when using the Buffer for this channel.

The built-in FIFO is available in this mode and can be configured in one data direction Transmit or Receive, then the FIFO is seen as a unique FFNBYTE bytes buffer, or it can also be configured in both data direction Transmit and Receive, then the FIFO is split into two separate FFNBYTE/2 bytes buffer with their own address space management, in this last case the definition of AEL and AFL levels is based on FFNBYTE bytes and is under Local Host responsibility.

23.3.4.4 Slave Receive-Only Mode

The slave receive mode is programmable (bits TRM set to 01 in the register (I)CONF).

In receive only mode, the Transmitter register should be loaded before McSPI is selected by an external SPI master device. Transmitter register or FIFO (if use of buffer enabled) content is always loaded in shift register whether it has been updated or not. The event TX_underflow is activated accordingly, and does not prevent transmission.

When an SPI word transfer completes (the SPIm.SPI_CH*n*STAT0[2] EOT bit (with *n* = 0) is set to 1), the received data is transferred to the channel receive register.

To use McSPI as a slave receive only device with MCSPI_CH(I)CONF[TRM]=00, the user has the responsibility of the inhibition of the TX_empty and TX_underflow interrupts and DMA write requests due to the transmitter register sate.

On completion of SPI word transfer (bit EOT of the register MCSPI_CH(I)STAT is set) the received data is transferred to the channel receive register. This bit is meaningless when using the Buffer for this channel. The built-in FIFO is available in this mode and can be configured with FFER bit field in the MCSPI_CH(I)CONF register, then the FIFO is seen as a unique FFNBYTE bytes buffer.

Figure 23-24 shows an example of a half-duplex system with a master device on the left and a receive-only slave device on the right. Each time a bit transfers out from the master, 1 bit transfers in from the slave. After eight cycles of the serial clock spim_clk, Word A transfers from the master to the slave.

Figure 23-24. SPI Half-Duplex Transmission (Receive-Only Slave)

Copyright © 2011, Texas Instruments Incorporated

23.3.4.5 Slave Transmit-Only Mode

The slave transmit only mode is programmable (bits TRM set to 10 in the register MCSPI_CH(I)CONF). This mode avoids the CPU to read the receiver register (minimizing data movement) when only transmission is meaningful.

To use McSPI as a slave transmit only device with (I)CONF[TRM]=00, the user should inhibit the RX_full and RX_overflow interrupts and DMA read requests due to the receiver register sate.

On completion of SPI word transfer the bit EOT of the register MCSPI_CH(I)STAT is set. This bit is meaningless when using the Buffer for this channel.

The built-in FIFO is available in this mode and can be configured with FFER bit field in the MCSPI_CH(I)CONF register, then the FIFO is seen as a unique FFNBYTE bytes buffer.

Figure 23-25 shows a half-duplex system with a master device on the left and a transmit-only slave device on the right. Each time a bit transfers out from the slave device, 1 bit transfers in the master. After eight cycles of the serial clock spim_clk, WordB transfers from the slave to the master.

Figure 23-25. SPI Half-Duplex Transmission (Transmit-Only Slave)

23.3.5 Interrupts

According to its transmitter register state and its receiver register state each channel can issue interrupt events if they are enabled.

The interrupt events are listed in the Section 23.3.3.2 and in Section 23.3.4.2.

Each interrupt event has a status bit, in the MCSPI_IRQSTATUS register, which indicates service is required, and an interrupt enable bit, in the MCSPI_IRQENABLE register, which enables the status to generate hardware interrupt requests.

When an interrupt occurs and later a mask is applied on it (IRQENABLE), the interrupt line is not asserted again even if the interrupt source has not been serviced.

McSPI supports interrupt driven operation and polling.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.3.5.1 Interrupt-Driven Operation

Alternatively, an interrupt enable bit, in the MCSPI_IRQENABLE register, can be set to enable each of the events to generate interrupt requests when the corresponding event occurs. Status bits are automatically set by hardware logic conditions.

When an event occurs (the single interrupt line is asserted), the CPU must:

- Read the MCSPI_IRQSTATUS register to identify which event occurred,
- Read the receiver register that corresponds to the event, to remove the source of an RX_full event, or Write into the transmitter register that corresponds to the event, to remove the source of a TX_empty event. No action is needed to remove the source of the events TX_underflow and RX_overflow.
- Write a 1 into the corresponding bit of MCSPI_IRQSTATUS register to clear the interrupt status, and release the interrupt line.

The interrupt status bit should always be reset after channel enabling and before event are enabled as interrupt source.

23.3.5.2 Polling

When the interrupt capability of an event is disabled in the MCSPI_IRQENABLE register, the interrupt line is not asserted and:

- The status bits in the MCSPI_IRQSTATUS register can be polled by software to detect when the corresponding event occurs.
- Once the expected event occurs, CPU must: Read the receiver register that corresponds to the event, to remove the source of an RX_full event, or write into the transmitter register that corresponds to the event, to remove the source of a TX_empty event. No action is needed to remove the source of the events TX_underflow and RX_overflow.
- Writing a 1 into the corresponding bit of MCSPI_IRQSTATUS register clears the interrupt status and does not affect the interrupt line state.

23.3.6 DMA Requests

McSPI can be interfaced with a DMA controller. At system level, the advantage is to discharge the local host of the data transfers.

According to its transmitter register state, its receiver register state or FIFO level (if use of buffer for the channel) each channel can issue DMA requests if they are enabled.

The DMA requests need to be disabled in order to get TX and RX interrupts, in order to define either the end of the transfer or the transfer of the last words for the modes listed below:

- Master Transmit On
- Master normal receive only mode
- Master turbo receive only m
- Slave Transmit Only

There are 2 DMA request lines per channel. The management of DMA requests differ according to use of FIFO buffer or not:

23.3.6.1 FIFO Buffer Disabled

The DMA Read request line is asserted when the channel is enabled and a new data is available in the receive register of the channel. DMA Read request can be individually masked with the bit DMAR of the register MCSPI_CH(I)CONF. The DMA Read request line is de-asserted (OCP compliant as defined in [1]) on read completion of the receive register of the channel.

The DMA Write request line is asserted when the channel is enabled and the transmitter register of the channel is empty. DMA Write request can be individually masked with the bit DMAW of the register MCSPI_CH(I)CONF. The DMA Write request line is deasserted (OCP compliant as defined in [1]) on load completion of the transmitter register of the channel.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Only one SPI word can be transmitted/received per OCP bus access to write/read the transmit or receive register.

23.3.6.2 FIFO Buffer Enabled

The DMA Read request line is asserted when the channel is enabled and a number of bytes defined in MCSPI_XFERLEVEL[AFL] bit field is hold in FIFO buffer for the receive register of the channel. DMA Read request can be individually masked with the bit DMAR of the register MCSPI_CH(I)CONF. The DMA Read request line is de-asserted (OCP compliant as defined in [1]) on the first SPI word read completion of the receive register of the channel. No new DMA request will be asserted again as soon as user has not performed the right number of read accesses defined by MCSPI_XFERLEVEL[AFL] it is under user responsibility.

The DMA Write request line is asserted when the channel is enabled and the number of bytes hold in FIFO buffer is below the level defined by the MCSPI_XFERLEVEL[AEL] bit field. DMA Write request can be individually masked with the bit DMAW of the register MCSPI_CH(I)CONF. The DMA Write request line is de-asserted (OCP compliant as defined in [1]) on load completion of the first SPI word in the transmitter register of the channel. No new DMA request will be asserted again as soon as user has not performed the right number of write accesses defined by MCSPI_XFERLEVEL[AEL] it is under user responsibility.

Only one SPI word can be transmitted/received per OCP bus access to write/read the transmit or receive FIFO.

23.3.6.3 DMA 256 Bit Aligned Addresses

The controller has two registers MCSPI_DAFTX and MCSPI_DAFRX only used with an enabled channel which manages the FIFO to be compliant the a DMA handler providing only 256 bit aligned addresses.

This features is activated when the bit field MCSPI_MODULCTRL[8] FDAA is set to ‘1’ and only one enabled channel have its bit field MCSPI_CH(I)CONF[27] FFE(I)W or MCSPI_CH(I)CONF[28] FFE(I)R enabled.

In this case the registers MCSPI_TX(I) and MCSPI_RX(I) are not used and datas are managed through registers MCSPI_DAFTX and MCSPI_DAFRX.

23.3.7 Emulation Mode

The MReqDebug input differentiates a regular access of a processor (application access), from an emulator access.

Application access: MReqDebug = 0

In functional mode, the consequences of a read of a receiver register MCSPI_RX(i) are the following:

- The source of an RXi_Full event in the MCSPI_IRQSTATUS register is removed, if it was enabled in the MCSPI_IRQENABLE register.
- The RXiS status bit in the MCSPI_IRQSTATUS register is cleared.
- In master mode, depending on the round robin arbitration, and the transmitter register state, the channel may access to the shift register for transmission/reception.

Emulator access: MReqDebug = 1

In emulation mode, McSPI behavior is the same as in functional mode but a read of a receiver register MCSPI_RX(i) is not intrusive:

- MCSPI_RX(i) is still considered as not read. When the FIFO buffer is enabled, pointers are not updated.
- The source of an RXi_Full event in the MCSPI_IRQSTATUS register is not removed. The RXiS status bit in the MCSPI_CH(i)STAT register is held steady.

In emulation mode, as in functional mode, based on the ongoing data transfers, the status bits of the MCSPI_CH(i)STAT register may be optionally updated, the interrupt and DMA request lines may be optionally asserted.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.3.8 Power Saving Management

Independently of the module operational modes (Transmit and/or Receive), two modes of operations are defined from a power management perspective: normal and idle modes.

The two modes are temporally fully exclusive.

McSPI is compliant with the profile “IdleReq / SIdleAck / Swakup” in the Idle mode defined in reference [4].

23.3.8.1 Normal Mode

Both the OCP clock and SPI clock (CLKSPIREF) provided to McSPI must be active for both master and slave modes. The auto-gating of the module OCP clock and SPI clock occurs when the following conditions are met:

- The bit AutoIdle of the register MCSPI_SYSCONFIG is set.
- In master mode, there is no data to transmit or receive in all channels.
- In slave mode, the SPI is not selected by the external SPI master device and no OCP accesses.

Autogating of the module OCP clock and SPI clock stops when the following conditions are met:

- In master mode, an OCP access occurs.
- In slave mode, an OCP access occurs or McSPI is selected by an external SPI master device.

23.3.8.2 Idle Mode

The OCP clock and SPI clock provided to McSPI may be switched off on system power manager request and switched back on module request.

McSPI is compliant with the power management handshaking protocol: idle request from the system power manager, idle acknowledgement from McSPI.

The idle acknowledgement in response to an idle request from the system power manager varies according to a programmable mode in the MCSPI_SYSCONFIG register: No idle mode, force idle mode, and smart idle mode.

- When programmed for no idle mode (the bit SIdleMode of the register MCSPI_SYSCONFIG is set to “01”), the module ignores the system power manager request, and behaves normally, as if the request was not asserted.
- When programmed for smart idle mode (the bit SIdleMode of the register MCSPI_SYSCONFIG is set to “10”), the module acknowledges the system power manager request according to its internal state.
- When programmed for force idle mode (the bit SIdleMode of the register MCSPI_SYSCONFIG is set to “00”), the module acknowledges the system power manager request unconditionally.

The OCP clock will be optionally switched off, during the smart idle mode period, if the bit ClockActivity of the register MCSPI_SYSCONFIG is set.

The SPI clock will be optionally switched off, during the smart idle mode period, if the second bit ClockActivity of the register MCSPI_SYSCONFIG is set.

McSPI assumes that both clocks may be switched off whatever the value set in the field ClockActivity of the register MCSPI_SYSCONFIG.

23.3.8.2.1 Transitions from Normal Mode to Smart-Idle Mode

The module detects an idle request when the synchronous signal IdleReq is asserted.

When IdleReq is asserted, any access to the module will generate an error as long as the OCP clock is alive.

When configured as a slave device, McSPI responds to the idle request by asserting the SIdleAck signal (idle acknowledgement) only after completion of the current transfer (SPIEN slave selection signal deasserted by the external master) and if interrupt or DMA request lines are not asserted.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

As a master device, McSPI responds to the idle request by asserting the SIdleAck signal (idle acknowledgement) only after completion of all the channel data transfers and if interrupt or DMA request lines are not asserted.

As long as SIdleAck is not asserted, if an event occurs, the module can still generate an interrupt or a DMA request after IdleReq assertion. In this case, the module ignores the idle request and SIdleAck will not get asserted: The system power manager will abort the power mode transition procedure. It is then the responsibility of the system to de-assert IdleReq before attempting to access the module.

When SIdleAck is asserted, the module does not assert any new interrupt or DMA request.

23.3.8.2.2 Transition From Smart-Idle Mode to Normal mode

McSPI detects the end of the idle period when the idle request signal (IdleReq) is deasserted.

Upon IdleReq de-assertion, the module switches back to normal mode and de-asserts SIdleAck signal. The module is fully operational.

23.3.8.2.3 Force-Idle Mode

Force-idle mode is enabled as follows:

- The bit SIdleMode of the register MCSPI_SYSCONFIG is cleared to “00” (Force Idle).

  The force idle mode is an idle mode where McSPI responds unconditionally to the idle request by asserting the SIdleAck signal and by deasserting unconditionally the interrupt and DMA request lines if asserted.

  The transition from normal mode to idle mode does not affect the interrupt event bits of the MCSPI_IRQSTATUS register.

  In force-idle mode, the module is supposed to be disabled at that time, so the interrupt and DMA request lines are likely deasserted. OCP clock and SPI clock provided to McSPI can be switched off.

  An idle request during an SPI data transfer can lead to an unexpected and unpredictable result, and is under software responsibility.

  Any access to the module in force idle mode will generate an error as long as the OCP clock is alive and IdleReq is asserted.

The module exits the force idle mode when:

- The idle request signal (IdleReq) is de-asserted.

  Upon IdleReq de-assertion, the module switches back to normal mode and de-asserts SIdleAck signal. The module is fully operational. The interrupt and DMA request lines are optionally asserted a clock cycle later.

23.3.9 System Test Mode

McSPI is in system test mode (SYSTEST) when the bit System_Test of the register MCSPI_MODULCTRL is set.

The SYSTEST mode is used to check in a very simple manner the correctness of the system interconnect either internally to interrupt handler, or power manager, or externally to SPI I/Os.

I/O verification can be performed in SYSTEST mode by toggling the outputs and capturing the logic state of the inputs. (See MCSPI_SYST register definition in Section 23.5.5)

23.3.10 Reset

23.3.10.1 Internal Reset Monitoring

The module is reset by the hardware when an active-low reset signal, synchronous to the OCP interface clock is asserted on the input pin RESETN.

This hardware reset signal has a global reset action on the module. All configuration registers and all state machines are reset, in all clock domains.

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Additionally, the module can be reset by software through the bit SoftReset of the register MCSPI_SYSCONFIG. This bit has exactly the same action on the module logic as the hardware RESETN signal. The register MCSPI_SYSCONFIG is not sensitive to software reset. The SoftReset control bit is active high. The bit is automatically reset to 0 by the hardware.

A global ResetDone status bit is provided in the status register MCSPI_SYSSTATUS. This bit is set to 1 when all the different clock domains resets (OCP domain and SPI domains) have been released (logical AND).

The global ResetDone status bit can be monitored by the software to check if the module is ready-to-use following a reset (either hardware or software).

The clock CLKSPIREF must be provided to the module, in order to allow the ResetDone status bit to be set.

When used in slave mode, the clock CLKSPIREF is needed only during the reset phase. The clock CLKSPIREF can be switched off after the ResetDone status is set.

23.3.10.2 Reset values of registers

The reset values of registers and signals are described in Section 23.5.

23.3.11 Access to Data Registers

This section details the supported data accesses (read or write) from/to the data receiver registers MCSPI_RX(i) and data transmitter registers MCSPI_TX(i).

Supported access:

McSPI supports only one SPI word per register (receiver or transmitter) and does not support successive 8-bit or 16-bit accesses for a single SPI word.

The SPI word received is always right justified on LSbit of the 32bit register MCSPI_RX(i), and the SPI word to transmit is always right justified on LSbit of the 32bit register MCSPI_TX(i).

The upper bits, above SPI word length, are ignored and the content of the data registers is not reset between the SPI data transfers.

The coherence between the number of bits of the SPI Word, the number of bits of the access and the enabled byte remains under the user's responsibility. Only aligned accesses are supported.

In Master mode, data should not be written in the transmit register when the channel is disbaled.

23.3.12 Programming Aid

23.3.12.1 Module Initialization

- Hard or soft reset.
- Read MCSPI_SYSSTATUS.
- Check if reset is done.
- Module configuration: (a) Write into MCSPI_MODULCTRL (b) Write into MCSPI_SYSCONFIG.
- Before the ResetDone bit is set, the clocks CLK and CLKSPIREF must be provided to the module.
- To avoid hazardous behavior, it is advised to reset the module before changing from MASTER mode to SLAVE mode or from SLAVE mode to MASTER mode.

23.3.12.2 Common Transfer Sequence

McSPI module allows the transfer of one or several words, according to different modes:

- MASTER, MASTER Turbo, MASTER LoSSI, SLAVE
- TRANSMIT - RECEIVE, TRANSMIT ONLY, RECEIVE ONLY
- Write and Read requests: Interrupts, DMA
- SPIEN lines assertion/deassertion: automatic, manual

Copyright © 2011, Texas Instruments Incorporated

For all these flows, the host process contains the main process and the interrupt routines. The interrupt routines are called on the interrupt signals or by an internal call if the module is used in polling mode.

In multi-channel master mode, the flows of different channels can be run simultaneously.

23.3.12.3 Main Program

- Interrupt Initialization: (a) Reset status bits in MCSPI_IRQSTATUS (b) Enable interrupts in MCSPI_IRQENA.
- Channel Configuration: Write MCSPI_CH(i)CONF.
- Start the channel: Write 0000 0001h in MCSPI_CH(i)CTRL.
- First write request: TX empty - Generate DMA write event/ polling TX empty flag by CPU to write First transmit word into MCSPI_TX(i).
- End of transfer: Stop the channel by writing 0000 0000h in MCSPI_CH(i)CTRL

The end of transfer depends on the transfer mode.

In multi-channel master mode, be careful not to overwrite the bits of other channels when initializing MCSPI_IRQSTATUS and MCSPI_IRQENABLE.

23.3.13 Interrupt and DMA Events

McSPI has two DMA requests (Rx and Tx) per channel. It also has one interrupt line for all the interrupt requests.

23.4 Use Cases

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.5 SPI Registers

Table 23-11 lists the McSPI registers.

Table 23-11. SPI Registers

Offset Address	Acronym	Register Name	Section
110h	MCSPI_SYSCONFIG	McSPI system configuration register	Section 23.5.1
114h	MCSPI_SYSSTATUS	McSPI system status register	Section 23.5.2
118h	MCSPI_IRQSTATUS	McSPI interrupt status register	Section 23.5.3
11Ch	MCSPI_IRQENABLE	McSPI interrupt enable register	Section 23.5.4
124h	MCSPI_SYST	McSPI system register	Section 23.5.5
128h	MCSPI_MODULCTRL	McSPI module control register	Section 23.5.6
12Ch	MCSPI_CH0CONF	McSPI channel *i* configuration register	Section 23.5.7
130h	MCSPI_CH0STAT	McSPI channel *i* status register	Section 23.5.8
134h	MCSPI_CH0CTRL	McSPI channel *i* control register	Section 23.5.9
138h	MCSPI_TX0	McSPI channel *i* FIFO transmit buffer register	Section 23.5.10
13Ch	MCSPI_RX0	McSPI channel *i* FIFO receive buffer register	Section 23.5.11
140h	MCSPI_CH1CONF	McSPI channel *i* configuration register	Section 23.5.7
144h	MCSPI_CH1STAT	McSPI channel *i* status register	Section 23.5.8
148h	MCSPI_CH1CTRL	McSPI channel *n* control register	Section 23.5.9
14Ch	MCSPI_TX1	McSPI channel *i* FIFO transmit buffer register	Section 23.5.10
150h	MCSPI_RX1	McSPI channel *i* FIFO receive buffer register	Section 23.5.11
154h	MCSPI_CH2CONF	McSPI channel *i* configuration register	Section 23.5.7
158h	MCSPI_CH2STAT	McSPI channel *i* status register	Section 23.5.8
15Ch	MCSPI_CH2CTRL	McSPI channel *i* control register	Section 23.5.9
160h	MCSPI_TX2	McSPI channel *i* FIFO transmit buffer register	Section 23.5.10
164h	MCSPI_RX2	McSPI channel *i* FIFO receive buffer register	Section 23.5.11
168h	MCSPI_CH3CONF	McSPI channel *i* configuration register	Section 23.5.7
16Ch	MCSPI_CH3STAT	McSPI channel *i* status register register	Section 23.5.8
170h	MCSPI_CH3CTRL	McSPI channel *i* control register	Section 23.5.9
174h	MCSPI_TX3	McSPI channel *i* FIFO transmit buffer register	Section 23.5.10
178h	MCSPI_RX3	McSPI channel *i* FIFO receive buffer register	Section 23.5.11
7Ch	MCSPI_XFERLEVEL	McSPI transfer levels register	Section 23.5.12

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.5.1 McSPI System Configuration Register (MCSPI_SYSCONFIG)

The McSPI system configuration register (MCSPI_SYSCONFIG) allows control of various parameters of the module interface. It is not sensitive to software reset. The MCSPI_SYSCONFIG is shown in Figure 23-26 and described in Table 23-12.

Figure 23-26. McSPI System Configuration Register (MCSPI_SYSCONFIG)

31						16
Reserved						
R/W-0						

15				10	9	8
Reserved					CLOCKACTIVITY	
R/W-0					R/W-0	

7	5	4	3	2	1	0
Reserved		SIDLEMODE		Reserved	SOFTRESET	AUTOIDLE
R/W-0		R/W-2h		R	R/W-0	R/W-1

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 23-12. McSPI System Configuration Register (MCSPI_SYSCONFIG) Field Descriptions

Bit	Field	Value	Description
31-10	Reserved	0	Reads returns 0
9-8	CLOCKACTIVITY		Clocks activity during wake-up mode period.
		0	OCP and Functional clocks may be switched off.
		1h	OCP clock is maintained. Functional clock may be switched-off.
		2h	Functional clock is maintained. OCP clock may be switched-off.
		3h	OCP and Functional clocks are maintained.
7-5	Reserved	0	Reads returns 0
4-3	SIDLEMODE		Power management
		0	If an idle request is detected, the McSPI acknowledges it unconditionally and goes in Inactive mode. Interrupt, DMA requests are unconditionally de-asserted.
		1h	If an idle request is detected, the request is ignored and keeps on behaving normally.
		2h	Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements.
		3h	Reserved
2	Reserved	0	Reserved
1	SOFTRESET		Software reset. During reads it always returns 0.
		0	(write) Normal mode
		1	(write) Set this bit to 1 to trigger a module reset. The bit is automatically reset by the hardware.
0	AUTOIDLE		Internal OCP Clock gating strategy
		0	OCP clock is free-running
		1	Automatic OCP clock gating strategy is applied, based on the OCP interface activity

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.5.2 McSPI System Status Register (MCSPI_SYSSTATUS)

The McSPI system status register (MCSPI_SYSSTATUS) provides status information about the module excluding the interrupt status information. The MCSPI_SYSSTATUS is shown in Figure 23-27 and described in Table 23-13.

Figure 23-27. McSPI System Status Register (MCSPI_SYSSTATUS)

31		16
Reserved		
R-0		

15	1	0
Reserved		RESETDONE
R-0		R-0

LEGEND: R = Read only; -*n* = value after reset

Table 23-13. McSPI System Status Register (MCSPI_SYSSTATUS) Field Descriptions

Bit	Field	Value	Description
31-1	Reserved	0	Reserved for module specific status information. Read returns 0.
0	RESETDONE		Internal Reset Monitoring
		0	Internal module reset is on-going
		1	Reset completed

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.5.3 McSPI Interrupt Status Register (MCSPI_IRQSTATUS)

The McSPI interrupt status register (MCSPI_IRQSTATUS) regroups all the status of the module internal events that can generate an interrupt. The MCSPI_IRQSTATUS is shown in Figure 23-28 and described in Table 23-14.

NOTE: In SYSTEST mode, the bits of this register have no meaning and always read 0.

Figure 23-28. McSPI Interrupt Status Register (MCSPI_IRQSTATUS)

31							18	17	16
Reserved								EOW	Rsvd
R/W-0								R/W-0	R-0

15	14	13	12	11	10	9	8
Rsvd	RX3_FULL	TX3_UNDERFLOW	TX3_EMPTY	Reserved	RX2_FULL	TX2_UNDERFLOW	TX2_EMPTY
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

7	6	5	4	3	2	1	0
Rsvd	RX1_FULL	TX1_UNDERFLOW	TX1_EMPTY	RX0_OVERFLOW	RX0_FULL	TX0_UNDERFLOW	TX0_EMPTY
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 23-14. McSPI Interrupt Status Register (MCSPI_IRQSTATUS) Field Descriptions

Bit	Field	Value	Description
31-18	Reserved	0	Reads returns 0
17	EOW		End of word (EOW) count event when a channel is enabled using the FIFO buffer and the channel has sent the number of McSPI words defined by the MCSPI_XFERLEVEL[WCNT].
		Write 0	Event status bit is unchanged.
		Read 0	Event false.
		Write 1	Event status bit is reset.
		Read 1	Event is pending.
16	Reserved	0	Reserved
15	Reserved	0	Reads returns 0
14	RX3_FULL		Receiver register is full or almost full. Only when Channel 3 is enabled. This bit indicate FIFO almost full status when built-in FIFO is used for receive register (MCSPI_CH3CONF[FFE3R] is set).
		Write 0	Event status bit is unchanged.
		Read 0	Event false.
		Write 1	Event status bit is reset.
		Read 1	Event is pending.
13	TX3_UNDERFLOW		Transmitter register underflow. Only when Channel 3 is enabled. The transmitter register is empty (not updated by Host or DMA with new data) before its time slot assignment.
			Exception: No TX_underflow event when no data has been loaded into the transmitter register since channel has been enabled.
		Write 0	Event status bit is unchanged.
		Read 0	Event false.
		Write 1	Event status bit is reset.
		Read 1	Event is pending.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 23-14. McSPI Interrupt Status Register (MCSPI_IRQSTATUS) Field Descriptions (continued)

Bit	Field	Value	Description
12	TX3_EMPTY		Transmitter register is empty or almost empty. This bit indicate FIFO almost full status when built-in FIFO is used for transmit register (MCSPI_CH3CONF[FFE3W] is set).
			Note: Enabling the channel automatically raises this event.
		Write 0	Event status bit is unchanged.
		Read 0	Event false.
		Write 1	Event status bit is reset.
		Read 1	Event is pending.
11	Reserved	0	Reads returns 0
10	RX2_FULL		Receiver register full or almost full. Channel 2 This bit indicate FIFO almost full status when built-in FIFO is used for receive register (MCSPI_CH3CONF[FFE2R] is set).
		Write 0	Event status bit is unchanged.
		Read 0	Event false.
		Write 1	Event status bit is reset.
		Read 1	Event is pending.
9	TX2_UNDERFLOW		Transmitter register underflow. Channel 2
		Write 0	Event status bit is unchanged.
		Read 0	Event false.
		Write 1	Event status bit is reset.
		Read 1	Event is pending.
8	TX2_EMPTY		Transmitter register empty or almost empty. Channel 2. This bit indicate FIFO almost full status when built-in FIFO is used for transmit register (MCSPI_CH3CONF[FFE2W] is set).
		Write 0	Event status bit is unchanged.
		Read 0	Event false.
		Write 1	Event status bit is reset.
		Read 1	Event is pending.
7	Reserved	0	Reads returns 0
6	RX1_FULL		Receiver register full or almost full. Channel 1. This bit indicate FIFO almost full status when built-in FIFO is use for receive register (MCSPI_CH3CONF[FFE1R] is set).
		Write 0	Event status bit is unchanged.
		Read 0	Event false.
		Write 1	Event status bit is reset.
		Read 1	Event is pending.
5	TX1_UNDERFLOW		Transmitter register underflow. Channel 1.
		Write 0	Event status bit is unchanged.
		Read 0	Event false.
		Write 1	Event status bit is reset.
		Read 1	Event is pending.
4	TX1_EMPTY		Transmitter register empty or almost empty. Channel 1. This bit indicate FIFO almost full status when built-in FIFO is use for transmit register (MCSPI_CH3CONF[FFE1W] is set).
		Write 0	Event status bit is unchanged.
		Read 0	Event false.
		Write 1	Event status bit is reset.
		Read 1	Event is pending.
3	RX0_OVERFLOW		Receiver register overflow (slave mode only). Channel 0.
		Write 0	Event status bit is unchanged.
		Read 0	Event false.
		Write 1	Event status bit is reset.
		Read 1	Event is pending.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 23-14. McSPI Interrupt Status Register (MCSPI_IRQSTATUS) Field Descriptions (continued)

Bit	Field	Value	Description
2	RX0_FULL		Receiver register full or almost full. Channel 0. Receiver register full or almost full. Channel 0
		Write 0	Event status bit is unchanged.
		Read 0	Event false.
		Write 1	Event status bit is reset.
		Read 1	Event is pending.
1	TX0_UNDERFLOW		Transmitter register underflow. Channel 0.
		Write 0	Event status bit is unchanged.
		Read 0	Event false.
		Write 1	Event status bit is reset.
		Read 1	Event is pending.
0	TX0_EMPTY		Transmitter register empty or almost empty. Channel 0. This bit indicate FIFO almost full status when built-in FIFO is use for transmit register (MCSPI_CH3CONF[FFE0W] is set).
		Write 0	Event status bit is unchanged.
		Read 0	Event false.
		Write 1	Event status bit is reset.
		Read 1	Event is pending.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.5.4 McSPI Interrupt Enable Register (MCSPI_IRQENABLE)

This McSPI interrupt enable register (MCSPI_IRQENABLE) enables/disables the module internal sources of interrupt, on an event-by-event basis. The MCSPI_IRQENABLE is shown in Figure 23-29 and described in Table 23-15.

Figure 23-29. McSPI Interrupt Enable Register (MCSPI_IRQENABLE)

31							18	17	16
Reserved								EOWKE	Rsvd
R/W-0								R/W-0	R-0

15	14	13	12	11	10	9	8
Rsvd	RX3_FULL_ ENABLE	TX3_UNDERFLOW_ ENABLE	TX3_EMPTY_ ENABLE	Reserved	RX2_FULL_ ENABLE	TX2_UNDERFLOW_ ENABLE	TX2_EMPTY_ ENABLE
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

7	6	5	4	3	2	1	0
Rsvd	RX1_FULL_ ENABLE	TX1_UNDERFLOW_ ENABLE	TX1_EMPTY_ ENABLE	RX0_OVERFLOW_ ENABLE	RX0_FULL_ ENABLE	TX0_UNDERFLOW_ ENABLE	TX0_EMPTY_ ENABLE
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 23-15. McSPI Interrupt Enable Register (MCSPI_IRQENABLE) Field Descriptions

Bit	Field	Value	Description
31-18	Reserved	0	Reads return 0
17	EOWKE		End of word count interrupt enable.
		0	Interrupt is disabled.
		1	Interrupt is enabled.
16	Reserved		Reserved
15	Reserved	0	Reads return 0
14	RX3_FULL_ ENABLE		MCSPI_RX3 receiver register full or almost full interrupt enable (channel 3).
		0	Interrupt is disabled.
		1	Interrupt is enabled.
13	TX3_UNDERFLOW_ ENABLE		MCSPI_TX3 transmitter register underflow interrupt enable (channel 3).
		0	Interrupt is disabled.
		1	Interrupt is enabled.
12	TX3_EMPTY_ ENABLE		MCSPI_TX3 transmitter register empty or almost empty interrupt enable (channel 3).
		0	Interrupt is disabled.
		1	Interrupt is enabled.
11	Reserved	0	Reads return 0
10	RX2_FULL_ ENABLE		MCSPI_RX2 receiver register full or almost full interrupt enable (channel 2).
		0	Interrupt is disabled.
		1	Interrupt is enabled.
9	TX2_UNDERFLOW_ ENABLE		MCSPI_TX2 transmitter register underflow interrupt enable (channel 2).
		0	Interrupt is disabled.
		1	Interrupt is enabled.
8	TX2_EMPTY_ ENABLE		MCSPI_TX2 transmitter register empty or almost empty interrupt enable (channel 2).
		0	Interrupt is disabled.
		1	Interrupt is enabled.
7	Reserved	0	Reads return 0

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 23-15. McSPI Interrupt Enable Register (MCSPI_IRQENABLE) Field Descriptions (continued)

Bit	Field	Value	Description
6	RX1_FULL_ ENABLE		MCSPI_RX1 receiver register full or almost full interrupt enable (channel 1)
		0	Interrupt is disabled.
		1	Interrupt is enabled.
5	TX1_UNDERFLOW_ ENABLE		MCSPI_TX1 transmitter register underflow interrupt enable (channel 1).
		0	Interrupt is disabled.
		1	Interrupt is enabled.
4	TX1_EMPTY_ ENABLE		MCSPI_TX1 transmitter register empty or almost empty interrupt enable (channel 1).
		0	Interrupt is disabled.
		1	Interrupt is enabled.
3	RX0_OVERFLOW_ ENABLE		MCSPI_RX0 receivier register overflow interrupt enable (channel 0).
		0	Interrupt is disabled.
		1	Interrupt is enabled.
2	RX0_FULL_ ENABLE		MCSPI_RX0 receiver register full or almost full interrupt enable (channel 0).
		0	Interrupt is disabled.
		1	Interrupt is enabled.
1	TX0_UNDERFLOW_ ENABLE		MCSPI_TX0 transmitter register underflow interrupt enable (channel 0).
		0	Interrupt is disabled.
		1	Interrupt is enabled.
0	TX0_EMPTY_ ENABLE		MCSPI_TX0 transmitter register empty or almost empty interrupt enable (channel 0).
		0	Interrupt is disabled.
		1	Interrupt is enabled.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.5.5 McSPI System Register (MCSPI_SYST)

This McSPI system register (MCSPI_SYST) is used to configure the system interconnect either internally to the peripheral bus or externally to the device I/O pads, when the module is configured in the system test (SYSTEST) mode. The MCSPI_SYST is shown in Figure 23-30 and described in Table 23-16.

Figure 23-30. McSPI System Register (MCSPI_SYST)

31							16
Reserved							
R/W-0							
15			12	11	10	9	8
Reserved				SSB	SPIENDIR	SPIDATDIR1	SPIDATDIR0
R/W-0				R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
Reserved	SPICLK	SPIDAT_1	SPIDAT_0	SPIEN_3	SPIEN_2	SPIEN_1	SPIEN_0
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 23-16. McSPI System Register (MCSPI_SYST) Field Descriptions

Bit	Field	Value	Description
31-12	Reserved	0	Reads returns 0
11	SSB		Set status bit. This bit must be cleared prior attempting to clear a status bit of the MCSPI_IRQSTATUS register.
		0	No action. Writing 0 does not clear already set status bits.
			This bit must be cleared prior attempting to clear a status bit of the <MCSPI_IRQSTATUS> register.
		1	Writing 1 sets to 1 all status bits contained in the MCSPI_IRQSTATUS register.
			Writing 1 into this bit sets to 1 all status bits contained in the <MCSPI_IRQSTATUS> register.
10	SPIENDIR		Sets the direction of the SPIEN[3:0] lines and SPICLK line.
		0	Output (as in master mode).
		1	Input (as in slave mode).
9	SPIDATDIR1		Sets the direction of the SPIDAT[1].
		0	Output
		1	Input
8	SPIDATDIR0		Sets the direction of the SPIDAT[0].
		0	Output
		1	Input
7	Reserved	0	Reserved
6	SPICLK		SPICLK line (signal data value)
			If MCSPI_SYST[SPIENDIR] = 1 (input mode direction), this bit returns the value on the CLKSPI line (high or low), and a write into this bit has no effect.
			If MCSPI_SYST[SPIENDIR] = 0 (output mode direction), the CLKSPI line is driven high or low according to the value written into this register.
5	SPIDAT_1		SPIDAT[1] line (signal data value)
			If MCSPI_SYST[SPIDATDIR1] = 0 (output mode direction), the SPIDAT[1] line is driven high or low according to the value written into this register.
			If MCSPI_SYST[SPIDATDIR1] = 1 (input mode direction), this bit returns the value on the SPIDAT[1] line (high or low), and a write into this bit has no effect.
4	SPIDAT_0		SPIDAT[0] line (signal data value)
			If MCSPI_SYST[SPIDATDIR0] = 0 (output mode direction), the SPIDAT[0] line is driven high or low according to the value written into this register.
			If MCSPI_SYST[SPIDATDIR0] = 1 (input mode direction), this bit returns the value on the SPIDAT[0] line (high or low), and a write into this bit has no effect.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 23-16. McSPI System Register (MCSPI_SYST) Field Descriptions (continued)

Bit	Field	Value	Description
3	SPIEN_3		SPIEN[3] line (signal data value) If MCSPI_SYST[SPIENDIR] = 0 (output mode direction), the SPIENT[3] line is driven high or low according to the value written into this register. If MCSPI_SYST[SPIENDIR] = 1 (input mode direction), this bit returns the value on the SPIEN[3] line (high or low), and a write into this bit has no effect.
2	SPIEN_2		SPIEN[2] line (signal data value) If MCSPI_SYST[SPIENDIR] = 0 (output mode direction), the SPIENT[2] line is driven high or low according to the value written into this register. If MCSPI_SYST[SPIENDIR] = 1 (input mode direction), this bit returns the value on the SPIEN[2] line (high or low), and a write into this bit has no effect.
1	SPIEN_1		SPIEN[1] line (signal data value) If MCSPI_SYST[SPIENDIR] = 0 (output mode direction), the SPIENT[1] line is driven high or low according to the value written into this register. If MCSPI_SYST[SPIENDIR] = 1 (input mode direction), this bit returns the value on the SPIEN[1] line (high or low), and a write into this bit has no effect.
0	SPIEN_0		SPIEN[0] line (signal data value) If MCSPI_SYST[SPIENDIR] = 0 (output mode direction), the SPIENT[0] line is driven high or low according to the value written into this register. If MCSPI_SYST[SPIENDIR] = 1 (input mode direction), this bit returns the value on the SPIEN[0] line (high or low), and a write into this bit has no effect.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.5.6 McSPI Module Control Register (MCSPI_MODULCTRL)

This McSPI module control register (MCSPI_MODULCTRL) is used to configure the serial port interface. The MCSPI_MODULCTRL is shown in Figure 23-31 and described in Table 23-17.

Figure 23-31. McSPI Module Control Register (MCSPI_MODULCTRL)

31															16
Reserved															
R/W-0															

15														9	8
Reserved															FDAA
R/W-0															R/W-0

7	6			4	3	2	1	0
MOA	INITDLY				SYSTEM_TEST	MS	PIN34	SINGLE
R/W-0	R/W-0				R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 23-17. McSPI Module Control Register(MCSPI_MODULCTRL) Field Descriptions

Bit	Field	Value	Description
31-9	Reserved	0	
8	FDAA		FIFO DMA Address 256-bit aligned. This register is used when a FIFO is managed by the module and DMA connected to the controller provides only 256 bit aligned address. If this bit is set the enabled channel which uses the FIFO has its datas managed through MCSPI_DAFTX and MCSPI_DAFRX registers instead of MCSPI_TX(i) and MCSPI_RX(i) registers.
		0	FIFO data managed by MCSPI_TX(i) and MCSPI_RX(i) registers.
		1	FIFO data managed by MCSPI_DAFTX and MCSPI_DAFRX registers.
7	MOA		Multiple word ocp access. This register can only be used when a channel is enabled using a FIFO. It allows the system to perform multiple SPI word access for a single 32-bit OCP word access. This is possible for WL < 16.
		0	Multiple word access disabled
		1	Multiple word access enabled with FIFO
6-4	INITDLY		Initial SPI delay for first transfer. This register is an option only available in SINGLE master mode, The controller waits for a delay to transmit the first SPI word after channel enabled and corresponding TX register filled. This Delay is based on SPI output frequency clock, No clock output provided to the boundary and chip select is not active in 4 pin mode within this period.
		0	No delay for first SPI transfer
		1h	The controller wait 4 SPI bus clock
		2h	The controller wait 8 SPI bus clock
		3h	The controller wait 16 SPI bus clock
		4h	The controller wait 32 SPI bus clock
		5h-Fh	Reserved
3	SYSTEM_TEST		Enables the system test mode
		0	Functional mode
		1	System test mode (SYSTEST)
2	MS		Master/ Slave
		0	Master - The module generates the SPICLK and SPIEN[3:0]
		1	Slave - The module receives the SPICLK and SPIEN[3:0]
1	PIN34		Pin mode selection. This register is used to configure the SPI pin mode, in master or slave mode. If asserted the controller only use SIMO,SOMI and SPICLK clock pin for SPI transfers.
		0	SPIEN is used as a chip select.
		1	SPIEN is not used.
			In this mode all related option to chip select have no meaning.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 23-17. McSPI Module Control Register(MCSPI_MODULCTRL) Field Descriptions (continued)

Bit	Field	Value	Description
0	SINGLE		Single channel / Multi Channel (master mode only)
		0	More than one channel will be used in master mode.
		1	Only one channel will be used in master mode. This bit must be set in Force SPIEN mode.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.5.7 McSPI Channel *(i)* Configuration Register (MCSPI_CH(i)CONF)

The McSPI channel *i* configuration register (MCSPI_CH(i)CONF) is used to configure channel *i*. The (MCSPI_CH(i)CONF) is shown in Figure 23-32 and described in Table 23-18.

Figure 23-32. McSPI Channel *(i)* Configuration Register (MCSPI_CH(i)CONF)

31	30	29	28	27	26	25	24
Reserved		CLKG	FFER	FFEW	TCS		SBPOL
R/W-0		R/W-0	R/W-0	R/W-0	R/W-0		R/W-0

23	22	21	20	19	18	17	16
SBE	SPIENSLV		FORCE	TURBO	IS	DPE1	DPE0
R/W-0	R/W-0		R/W-0	R/W-0	R/W-1	R/W-1	R/W-0

15	14	13	12	11			8
DMAR	DMAW	TRM		WL			
R/W-0	R/W-0	R/W-0		R/W-0			

7	6	5			2	1	0
WL	EPOL	CLKD				POL	PHA
R/W-0	R/W-0	R/W-0				R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 23-18. McSPI Channel *(i)* Configuration Register (MCSPI_CH(i)CONF) Field Descriptions

Bit	Field	Value	Description
31-30	Reserved	0	Read returns 0
29	CLKG		Clock divider granularity. This register defines the granularity of channel clock divider: power of two or one clock cycle granularity. When this bit is set the register MCSPI_CHCTRL[EXTCLK] must be configured to reach a maximum of 4096 clock divider ratio. Then The clock divider ratio is a concatenation of MCSPI_CHCONF[CLKD] and MCSPI_CHCTRL[EXTCLK] values.
		0	Clock granularity of power of 2
		1	1 clock cycle granularity
28	FFER		FIFO enabled for receive. Only one channel can have this bit set.
		0	The buffer is not used to receive data.
		1	The buffer is used to receive data.
27	FFEW		FIFO enabled for transmit. Only one channel can have this bit set.
		0	The buffer is not used to transmit data.
		1	The buffer is used to transmit data.
26-25	TCS		Chip select time control. This 2-bits field defines the number of interface clock cycles between CS toggling and first or last edge of SPI clock.
		0	0.5 clock cycles
		1h	1.5 clock cycles
		2h	2.5 clock cycles
		3h	3.5 clock cycles
24	SBPOL		Start bit polarity.
		0	Start bit polarity is held to 0 during SPI transfer.
		1	Start bit polarity is held to 1 during SPI transfer.
23	SBE		Start bit enable for SPI transfer.
		0	Default SPI transfer length as specified by WL bit field.
		1	Start bit D/CX added before SPI transfer. Polarity is defined by MCSPI_CH(i)CONF[SBPOL].

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 23-18. McSPI Channel *(i)* Configuration Register (MCSPI_CH(i)CONF) Field Descriptions (continued)

Bit	Field	Value	Description
22-21	SPIENSLV		Channel 0 only and slave mode only: SPI slave select signal detection. Reserved bits (read returns 0) for other cases.
		0	Detection enabled only on SPIEN[0]
		1h	Detection enabled only on SPIEN[1]
		2h	Detection enabled only on SPIEN[2]
		3h	Detection enabled only on SPIEN[3]
20	FORCE		Manual SPIEN assertion to keep SPIEN active between SPI words. (single channel master mode only)
		0	Writing 0 into this bit drives the SPIEN line when MCSPI_CHCONF(i)[EPOL]=0, and drives it high when MCSPI_CHCONF(i)[EPOL]=1.
		1	Writing 1 into this bit drives the SPIEN line when MCSPI_CHCONF(i)[EPOL]=0, and drives it low when MCSPI_CHCONF(i)[EPOL]=1
19	TURBO		Turbo mode.
		0	Turbo is deactivated (recommended for single SPI word transfer).
		1	Turbo is activated to maximize the throughput for multi-SPI word transfers.
18	IS		Input select
		0	Data line 0 (SPIDAT[0]) selected for reception.
		1	Data line 1 (SPIDAT[1]) selected for reception.
17	DPE1		Transmission enable for data line 1 (SPIDATAGZEN[1])
		0	Data line 1 (SPIDAT[1]) selected for transmission
		1	No transmission on data line 1 (SPIDAT[1])
16	DPE0		Transmission enable for data line 0 (SPIDATAGZEN[0])
		0	Data line 0 (SPIDAT[0]) selected for transmission
		1	No transmission on data line 0 (SPIDAT[0])
15	DMAR		DMA read request. The DMA read request line is asserted when the channel is enabled and new data is available in the receive register of the channel. The DMA read request line is deasserted on read completion of the receive register of the channel.
		0	DMA read request is disabled.
		1	DMA read request is enabled.
14	DMAW		DMA write request. The DMA write request line is asserted when the channel is enabled and the MCSPI_TX*n* register of the channel is empty. The DMA write request line is deasserted on load completion of the MCSPI_TX*n* register of the channel.
		0	DMA write request is disabled.
		1	DMA write request is enabled.
13-12	TRM		Transmit/receive modes.
		0	Transmit and receive mode
		1h	Receive-only mode
		2h	Transmit-only mode
		3h	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 23-18. McSPI Channel *(i)* Configuration Register (MCSPI_CH(i)CONF) Field Descriptions (continued)

Bit	Field	Value	Description
11-7	WL		SPI word length.
		0	Reserved
		1h	Reserved
		2h	Reserved
		3h	The SPI word is 4-bits long.
		4h	The SPI word is 5-bits long
		5h	The SPI word is 6-bits long
		6h	The SPI word is 7-bits long
		7h	The SPI word is 8-bits long
		8h	The SPI word is 9-bits long
		9h	The SPI word is 10-bits long
		Ah	The SPI word is 11-bits long
		Bh	The SPI word is 12-bits long
		Ch	The SPI word is 13-bits long
		Dh	The SPI word is 14-bits long
		Eh	The SPI word is 15-bits long
		Fh	The SPI word is 16-bits long
		10h	The SPI word is 17-bits long
		11h	The SPI word is 18-bits long
		12h	The SPI word is 19-bits long
		13h	The SPI word is 20-bits long
		14h	The SPI word is 21-bits long
		15h	The SPI word is 22-bits long
		16h	The SPI word is 23-bits long
		17h	The SPI word is 24-bits long
		18h	The SPI word is 25-bits long
		19h	The SPI word is 26-bits long
		1Ah	The SPI word is 27-bits long
		1Bh	The SPI word is 28-bits long
		1Ch	The SPI word is 29-bits long
		1Dh	The SPI word is 30-bits long
		1Eh	The SPI word is 31-bits long
		1Fh	The SPI word is 32-bits long
6	EPOL		SPIEN polarity
		0	SPIEN is held high during the active state.
		1	SPIEN is held low during the active state.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 23-18. McSPI Channel *(i)* Configuration Register (MCSPI_CH(i)CONF) Field Descriptions (continued)

Bit	Field	Value	Description
5-2	CLKD		Frequency divider for SPICLK. (only when the module is a Master SPI device). A programmable clock divider divides the SPI reference clock (CLKSPIREF) with a 4-bit value, and results in a new clock SPICLK available to shift-in and shift-out data. By default the clock divider ratio has a power of two granularity when MCSPI_CHCONF[CLKG] is cleared, Otherwise this register is the 4 LSB bit of a 12-bit register concatenated with clock divider extension MCSPI_CHCTRL[EXTCLK] register.
			The value description below defines the clock ratio when MCSPI_CHCONF[CLKG] is cleared to 0.
		0	Divide by 1.
		1h	2
		2h	4
		3h	8
		4h	16
		5h	32
		6h	64
		7h	128
		8h	256
		9h	512
		Ah	1024
		Bh	2048
		Ch	4096
		Dh	8192
		Eh	16384
		Fh	32768
1	POL		SPICLK polarity
		0	SPICLK is held high during the active state
		1	SPICLK is held low during the active state
0	PHA		SPICLK phase
		0	Data are latched on odd numbered edges of SPICLK
		1	Data are latched on even numbered edges of SPICLK

Table 23-19. Data Lines Configurations

			TRMi		
ISi	DPEi1	DPEi0	Transmit and Receive	Receive Only	Transmit Only
0	0	0	Supported	Supported	Supported
0	0	1	Supported	Supported	Supported
0	1	0	Supported	Supported	Supported
0	1	1	Not supported (unpredictable result)	Supported	Not supported (unpredictable result)
1	0	0	Supported	Supported	Supported
1	0	1	Supported	Supported	Supported
1	1	0	Supported	Supported	Supported
1	1	1	Not supported (unpredictable result)	Supported	Not supported (unpredictable result)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.5.8 McSPI Channel (*i*) Status Register (MCSPI_CH(i)STAT)

The McSPI channel *i* status register register (MCSPI_CH(i)STAT) provides status information about the McSPI channel *i* FIFO transmit buffer register (MCSPI_TX*n*) and the McSPI channel *i* FIFO receive buffer register (MCSPI_RX*n*) of channel *i*. The (MCSPI_CH(i)STAT) is shown in Figure 23-33 and described in Table 23-20.

Figure 23-33. McSPI Channel *(i)* Status Register (MCSPI_CH(i)STAT)

31							8
Reserved							
R-0							
7	6	5	4	3	2	1	0
Reserved	RXFFF	RXFFE	TXFFF	TXFFE	EOT	TXS	RXS
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R = Read only; -*n* = value after reset

Table 23-20. McSPI Channel *(i)* Status Register (MCSPI_CH(i)STAT) Field Descriptions

Bit	Field	Value	Description
31-7	Reserved	0	Read returns 0
6	RXFFF		Channel *i* FIFO receive buffer full status.
		0	FIFO receive buffer is not full.
		1	FIFO receive buffer is full.
5	RXFFE		Channel *i* FIFO receive buffer empty status.
		0	FIFO receive buffer is not empty.
		1	FIFO receive buffer is empty.
4	TXFFF		Channel *i* FIFO transmit buffer full status.
		0	FIFO transmit buffer is not full.
		1	FIFO transmit buffer is full.
3	TXFFE		Channel *i* FIFO transmit buffer empty status.
		0	FIFO transmit buffer is not empty.
		1	FIFO transmit buffer is empty.
2	EOT		Channel *i* end-of-transfer status. The definitions of beginning and end of transfer vary with master versus slave and the transfer format (transmit/receive mode, turbo mode).
		0	This flag is automatically cleared when the shift register is loaded with the data from the transmitter register (beginning of transfer).
		1	This flag is automatically set to one at the end of an SPI transfer.
1	TXS		Channel *i* transmitter register status. The bit is cleared when the host writes the most significant byte of the SPI word in the MCSPI_TX(i) register. The bit is set when enabling the channel *i* , and also when the SPI word is transferred from the MCSPI_TX(i) register to the shift register.
		0	Register is full.
		1	Register is empty.
0	RXS		Channel *i* receiver register status. The bit is cleared when enabling the channel *i*, and also when the host reads the most significant byte of the received SPI word from the MCSPI_RX(i) register. The bit is set when the received SPI word is transferred from the shift register to the MCSPI_RX(i) register.
		0	Register is empty.
		1	Register is full.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.5.9 McSPI Channel (*i*) Control Register (MCSPI_CH(I)CTRL)

The McSPI channel *i* control register (MCSPI_CH*i*CTRL) is used to enable channel *i*. The MCSPI_CH*i*CTRL is shown in Figure 23-34 and described in Table 23-21.

Figure 23-34. McSPI Channel (*i*) Control Register (MCSPI_CH(I)CTRL)

31		16
Reserved		
R/W-0		

15	8	7	1	0
EXTCLK		Reserved		EN
R/W-0		R/W-0		R/W-0

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 23-21. McSPI Channel (*i*) Control Register (MCSPI_CH(I)CTRL) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15-8	EXTCLK		Clock ratio extension. Used to concatenate with the CLKD bit field in MCSPI_CH*n*CONF for clock ratio only when granularity is 1 clock cycle (CLKG bit in MCSPI_CH*n*CONF set to 1). Then the maximum value reached is a 4096 clock divider ratio.
		0	Clock ratio is CLKD + 1
		1h	Clock ratio is CLKD + 1 + 16
		...	...
		FFh	Clock ratio is CLKD + 1 + 4080
7-1	Reserved	0	Reserved
0	EN		Channel *n* enable.
		0	Channel *n* is not active.
		1	Channel *n* is active.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.5.10 McSPI Channel *(i)* Transmit Register (MCSPI_TX*(i)*)

The McSPI channel *i* transmit register (MCSPI_TX*(i)*) contains a single McSPI word to transmit on the serial link. The (MCSPI_TX*(i)*) is shown in Figure 23-35 and described in Table 23-22.

- Little endian host access SPI 8 bit word on 0; big endian host accesses on 3h.
- The SPI words are transferred with MSB first.

Figure 23-35. McSPI Channel *(i)* Transmit Register (MCSPI_TX*(i)*)

31	0
TDATA	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 23-22. McSPI Channel *(i)* Transmit Register (MCSPI_TX*(i)*) Field Descriptions

Bit	Field	Value	Description
31-0	TDATA	0-FFFF FFFFh	Channel *i* data to transmit.

23.5.11 McSPI Channel *(i)* Receive Register (MCSPI_RX*(i)*)

The McSPI channel *i* FIFO receive buffer register (MCSPI_RX*(i)*) contains a single McSPI word received through the serial link. The (MCSPI_RX*(i)*) is shown in Figure 23-36 and described in Table 23-23.

- Little endian host access SPI 8 bit word on 0; big endian host accesses on 3h.

Figure 23-36. McSPI Channel *(i)* Receive Register (MCSPI_RX*(i)*)

31	0
RDATA	
R-0	

LEGEND: R = Read only; -*n* = value after reset

Table 23-23. McSPI Channel *(i)* Receive Register (MCSPI_RX*(i)*) Field Descriptions

Bit	Field	Value	Description
31-0	RDATA	0-FFFF FFFFh	Channel *i* received data.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

23.5.12 McSPI Transfer Levels Register (MCSPI_XFERLEVEL)

The McSPI transfer levels register (MCSPI_XFERLEVEL) provides the transfer levels needed while using the FIFO buffer during transfer. The MCSPI_XFERLEVEL is shown in Figure 23-37 and described in Table 23-24.

Figure 23-37. McSPI Transfer Levels Register (MCSPI_XFERLEVEL)

31			16
WCNT			
R/W-0			

15	8	7	0
AFL		AEL	
R/W-0		R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 23-24. McSPI Transfer Levels Register (MCSPI_XFERLEVEL) Field Descriptions

Bit	Field	Value	Description
31-16	WCNT		SPI word counter. Holds the programmable value of the number of SPI words to be transferred on the channel that is using the FIFO buffer. When the transfer has started, a read back of this register returns the current SPI word transfer index.
		0	Counter not used
		1	1 SPI word
		...	...
		FFFEh	65534 SPI word
		FFFFh	65535 SPI word
15-8	AFL		Buffer almost full. Holds the programmable almost full level value used to determine almost full buffer condition. If you want an interrupt or a DMA read request to be issued during a receive operation when the data buffer holds at least *n* bytes, then the buffer MCSPI_MODULCTRL[AFL] must be set with n - 1.
		0	1 byte
		1	2 bytes
		...	...
		FFh	256 bytes
7-0	AEL		Buffer almost empty. Holds the programmable almost empty level value used to determine almost empty buffer condition. If you want an interrupt or a DMA write request to be issued during a transmit operation when the data buffer is able to receive *n* bytes, then the buffer MCSPI_MODULCTRL[AEL] must be set with n - 1.
		0	1 byte
		1	2 bytes
		...	...
		FFh	256 bytes

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Chapter 24

SPRUH73–July 2011

General-Purpose Input/Output

This chapter describes the GPIO of the device.

Topic **Page**

24.1 Introduction **2916**
24.2 Integration **2917**
24.3 Functional Description **2920**
24.4 GPIO Registers **2930**

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

24.1 Introduction

24.1.1 Purpose of the Peripheral

The general-purpose interface combines four general-purpose input/output (GPIO) modules. Each GPIO module provides 32 dedicated general-purpose pins with input and output capabilities; thus, the general-purpose interface supports up to 128 (4 × 32) pins. These pins can be configured for the following applications:

- Data input (capture)/output (drive)
- Keyboard interface with a debounce cell
- Interrupt generation in active mode upon the detection of external events. Detected events are processed by two parallel independent interrupt-generation submodules to support biprocessor operations.
- Wake-up request generation in idle mode upon the detection of external events.

24.1.2 GPIO Features

Each GPIO module is made up of 32 identical channels. Each channel can be configured to be used in the following applications:

- Data input/output
- Keyboard interface with a de-bouncing cell
- Synchronous interrupt generation (in active mode) upon the detection of external events (signal transition(s) and/or signal level(s))
- Wake-up request generation (in Idle mode) upon the detection of signal transition(s)

Global features of the GPIO interface are:

- Synchronous interrupt requests from each channel are processed by two identical interrupt generation sub-modules to be used independently by the Host ARM, Ducati, and the GEM
- Wake-up requests from input channels are merged together to issue one wake-up signal to the system
- Shared registers can be accessed through “Set & Clear” protocol

24.1.3 Unsupported GPIO Features

The wake-up feature of the GPIO modules is only supported on GPIO0.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

24.2 Integration

The device instantiates four GPIO_V2 modules. Each GPIO module provides the support for 32 dedicated pins with input and output configuration capabilities. Input signals can be used to generate interruptions and wake-up signal. Two Interrupt lines are available for bi-processor operation. Pins can be dedicated to be used as a keyboard controller.

With four GPIO modules, the device allows for a maximum of 128 GPIO pins. (The exact number available varies as a function of the device configuration and pin muxing.) GPIO0 is in the Wakeup domain and may be used to wakeup the device via external sources. GPIO[1:3] are located in the peripheral domain.

Figure 24-1 and Figure 24-2 show the GPIO integration.

Figure 24-1. GPIO0 Module Integration

Figure 24-2. GPIO[1–3] Module Integration

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

24.2.1 GPIO Connectivity Attributes

The general connectivity attributes for the GPIO modules in the device are shown in Table 24-1 and Table 24-2.

Table 24-1. GPIO0 Connectivity Attributes

Attributes	Type
Power Domain	Wakeup Domain
Clock Domain	PD_WKUP_L4_WKUP_GCLK (OCP) GPIO_0_GDBCLK (Debounce)
Reset Signals	WKUP_DOM_RST_N
Idle/Wakeup Signals	Smart Idle / Slave Wakeup
Interrupt Requests	Two Interrupts: INTRPEND1 to MPU subsystem, PRUSS and WakeM3 INTRPEND2 to MPU subsystem and WakeM3
DMA Requests	None
Physical Address	L4 Wakeup slave port

Table 24-2. GPIO[1:3] Connectivity Attributes

Attributes	Type
Power Domain	Peripheral Domain
Clock Domain	PD_PER_L4LS_GCLK (OCP) GPIO_1_GDBCLK (GPIO1 Debounce) GPIO_2_GDBCLK (GPIO2 Debounce) GPIO_3_GDBCLK (GPIO3 Debounce)
Reset Signals	PER_DOM_RST_N
Idle/Wakeup Signals	Smart Idle
Interrupt Requests	Two Interrupts: INTRPEND1 to MPU subsystem INTRPEND2 to MPU subsystem
DMA Requests	None
Physical Address	L4 Peripheral slave port

24.2.2 GPIO Clock and Reset Management

The GPIO modules require two clocks: The de-bounce clock is used for the de-bouncing cells. The interface clock provided by the peripheral bus (L4 interface) is also the functional clock and is used through the entire GPIO module (except within the de-bouncing sub-module). It clocks the OCP interface and the internal logic. For GPIO0 the debounce clock is selected from one of three sources using the CLKSEL_GPIO0_DBCLK register in the PRCM:

- The on-chip ~32.768 KHz oscillator (CLK_RC32K)
- The PER PLL generated 32.768 KHz clock (CLK_32KHZ)
- The external 32.768 KHz oscillator/clock (CLK_32K_RTC)

Table 24-3. GPIO Clock Signals

Clock Signal	Max Freq	Reference / Source	Comments
GPIO0			
Functional / Interface clock	100 MHz	CORE_CLKOUTM4 / 2	pd_wkup_l4_wkup_gclk From PRCM

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 24-3. GPIO Clock Signals (continued)

Clock Signal	Max Freq	Reference / Source	Comments
Debounce Functional clock	32.768 KHz	CLK_RC32K CLK_32KHZ (PER_CLKOUTM2 / 5859.375) CLK_32K_RTC	pd_wkup_gpio0_gdbclk From PRCM
GPIO[1:3]			
Functional / Interface clock	100 MHz	CORE_CLKOUTM4 / 2	pd_per_l4ls_gclk From PRCM
Debounce Functional clock (GPIO1)	32.768 KHz	CLK_32KHZ (PER_CLKOUTM2 / 5859.375)	pd_per_gpio_1_gdbclk From PRCM
Debounce Functional clock (GPIO2)	32.768 KHz	CLK_32KHZ (PER_CLKOUTM2 / 5859.375)	pd_per_gpio_2_gdbclk From PRCM
Debounce Functional clock (GPIO3)	32.768 KHz	CLK_32KHZ (PER_CLKOUTM2 / 5859.375)	pd_per_gpio_3_gdbclk From PRCM

24.2.3 GPIO Pin List

Each GPIO module includes 32 interface I/Os. These signals are designated as shown in Table 24-4. Note that for this device, most of these signals will be multiplexed with functional signals from other interfaces.

Table 24-4. GPIO Pin List

Pin	Type	Description
GPIO0		
GPIO0_[31:0]	I/O	General Purpose Input-Output pins
GPIO1		
GPIO1_[31:0]	I/O	General Purpose Input-Output pins
GPIO2		
GPIO2_[31:0]	I/O	General Purpose Input-Output pins
GPIO3		
GPIO3_[31:0]	I/O	General Purpose Input-Output pins

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

24.3 Functional Description

This section discusses the operational details and basic functions of the GPIO peripheral.

24.3.1 Operating Modes

Four operating modes are defined for the module:

- **Active mode:** the module is running synchronously on the interface clock, interrupt can be generated according to the configuration and the external signals.
- **Idle mode:** the module is in a waiting state, interface clock can be stopped , no interrupt can be generated, a wake-up signal can be generated according to the configuration and external signals. Check the chip top-level functional specification for the availability of the debouncing clock while in Idle mode. If the debouncing clock is active, the debouncing cell can be used to sample and to filter the input to generate a wakeup event. Otherwise (debouncing clock inactive), the debouncing cell cannot be used, as it would gate all input signals.
- **Inactive mode:** the module has no activity, interface clock can be stopped, no interrupt can be generated, and the wake-up feature is inhibited.
- **Disabled mode:** the module is not used, internal clock paths are gated, no interrupt or wake-up request can be generated.

The Idle and Inactive modes are configured within the module and activated on request by the host processor through system interface sideband signals. The Disabled mode is set by software through a dedicated configuration bit. It unconditionally gates the internal clock paths not use for the system interface. All module registers are 8, 16 or 32-bit accessible through the OCP compatible interface (little endian encoding). In active mode, the event detection (level or transition) is performed in the GPIO module using the interface clock. The detection's precision is set by the frequency of this clock and the selected internal gating scheme.

24.3.2 Clocking and Reset Strategy

24.3.2.1 Clocks

GPIO module runs using two clocks:

- The debouncing clock is used for the debouncing sub-module logic (without the corresponding configuration registers). This module can sample the input line and filters the input level using a programmed delay.
- The interface clock provided by the peripheral bus (OCP compatible system interface). It is used through the entire GPIO module (except within the debouncing sub-module logic). It clocks the OCP interface and the internal logic. Clock gating features allow adapting the module power consumption to the activity.

24.3.2.2 Clocks, Gating and Active Edge Definitions

The interface clock provided by the peripheral bus (OCP compatible system interface) is used through the entire GPIO module. Two clock domains are defined: the OCP interface and the internal logic. Each clock domain can be controlled independently. Sampling operations for the data capture and for the events detection are done using the rising edge. The data loaded in the data output register (GPIO_DATAOUT) is set at the output GPIO pins synchronously with the rising edge of the interface clock.

Five clock gating features are available:

- Clock for the system interface logic can be gated when the module is not accessed, if the AUTOIDLE configuration bit in the system configuration register (GPIO_SYSCONFIG) is set. Otherwise, this logic is free running on the interface clock.
- Clock for the input data sample logic can be gated when the data in register is not accessed.
- Four clock groups are used for the logic in the synchronous events detection. Each 8 input GPIO_V2 pins group will have a separate enable signal depending on the edge/level detection register setting. If a group requires no detection, then the corresponding clock will be gated. All channels are also gated using a 'one out of N' scheme. N can take the values 1, 2, 4 or 8. The interface clock is enabled for

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

this logic one cycle every N cycles. When N is equal to 1, there is no gating and this logic is free running on the interface clock. When N is between 2 to 8, this logic is running at the equivalent frequency of interface clock frequency divided by N.

- In Inactive mode, all internal clock paths are gated.
- In Disabled mode, all internal clock paths not used for the system interface are gated. All GPIO registers are accessible synchronously with the interface clock.

24.3.2.3 Sleep Mode Request and Acknowledge

Upon a Sleep mode request issued by the host processor, the GPIO module goes to the Idle mode according to the IDLEMODE field in the system configuration register (GPIO_SYSCONFIG).

- IDLEMODE = 0 (Force-Idle mode): the GPIO goes in Inactive mode independently of the internal module state and the Idle acknowledge is unconditionally sent. In Force-Idle mode, the module is in Inactive mode and its wake-up feature is totally inhibited.
- IDLEMODE = 1h (No-Idle mode): the GPIO does not go to the Idle mode and the Idle acknowledge is never sent.
- IDLEMODE = 2h (Smart-Idle mode) or IDLEMODE = 3h (Smart-Idle mode): the GPIO module evaluates its internal capability to have the interface clock switched off. Once there is no more internal activity (the data input register completed to capture the input GPIO pins, there is no pending interrupt, all interrupt status bits are cleared, and there is no write access to GPIO_DEBOUNCINGTIME register pending to be synchronized), the Idle acknowledge is asserted and the GPIO enters Idle mode, ready to issue a wake-up request when the expected transition occurs on an enabled GPIO input pin. This wake-up request is effectively sent only if the ENAWAKEUP bit in GPIO_SYSCONFIG is set to enable the GPIO wakeup capability. When the system is awake, the Idle Request goes inactive, the Idle acknowledge and wake-up request (if it is the GPIO that triggered the system's wakeup) signals are immediately de-asserted, and the asynchronous wake-up request (if existing) is reflected into the synchronous interrupt status registers.

NOTE: Idle mode request and Idle acknowledge are system interface sideband signals. Once the GPIO acknowledges the Sleep mode request (Idle acknowledge has been sent), the interface clock can be stopped anytime.

Upon a Sleep mode request issued by the host processor, the GPIO module goes to the Idle mode only if there is no active bit in GPIO_IRQSTATUS_RAW_n registers.

24.3.2.4 Reset

The OCP hardware Reset signal has a global reset action on the GPIO. All configuration registers, all DFFs clocked with the Interface clock or Debouncing clock and all internal state machines are reset when the OCP hardware Reset is active (low level). The RESETDONE bit in the system status register (GPIO_SYSSTATUS) monitors the internal reset status: it is set when the Reset is completed on both OCP and Debouncing clock domains. The software Reset (SOFTRESET bit in the system configuration register) has the same effect as the OCP hardware Reset signal, and the RESETDONE bit in GPIO_SYSSTATUS is updated in the same condition.

24.3.3 Interrupt and Wake-up Features

24.3.3.1 Functional Description

In order to generate an interrupt request to a host processor upon a defined event (level or logic transition) occurring on a GPIO pin, the GPIO configuration registers have to be programmed as follows:

- Interrupts for the GPIO channel must be enabled in the GPIO_IRQSTATUS_SET_0 and/or GPIO_IRQSTATUS_SET_1 registers.
- The expected event(s) on input GPIO to trigger the interrupt request has to be selected in the GPIO_LEVELDETECT0, GPIO_LEVELDETECT1, GPIO_RISINGDETECT, and GPIO_FALLINGDETECT registers.

Copyright © 2011, Texas Instruments Incorporated

In order to generate a wake-up request to a host processor upon a defined event (logic transition) occurring on a GPIO pin, the GPIO configuration registers have to be programmed as follows:

- The GPIO channel must be enabled in the GPIO_IRQWAKEN register.
- The expected event(s) on input GPIO to trigger the interrupt (or the wake-up) request has to be selected in the GPIO_RISINGDETECT and GPIO_FALLINGDETECT registers. Wake-up request can only be generated on rising and/or on falling transitions.

For instance, interrupt generation on both edges on input k is configured by setting to 1 the *k*th bit in registers GPIO_RISINGDETECT and GPIO_FALLINGDETECT along with the interrupt enabling for one or both interrupt lines (GPIO_IRQSTATUS_SET_n).

NOTE: All interrupt (or wake-up) sources (the 32 input GPIO channels) are merged together to issue two synchronous interrupt requests 1 and 2, and two asynchronous wake-up requests.

24.3.3.2 Synchronous Path: Interrupt Request Generation

In Active mode, once the GPIO configuration registers have been set to enable the interrupt generation, a synchronous path (Figure 24-3) samples the transitions and levels on the input GPIO with the internally gated interface clock. When an event matches the programmed settings, the corresponding bit in the GPIO_IRQSTATUS_RAW_n registers is set to 1 and, on the following interface clock cycle, the interrupt lines 1 and/or 2 are activated (depending on the GPIO_IRQSTATUS_SET_n registers).

Due to the sampling operation, the minimum pulse width on the input GPIO to trigger a synchronous interrupt request is two times the internally gated interface clock period (the internally gated interface clock period is equal to N times the interface clock period). This minimum pulse width has to be met before and after any expected level transition detection. Level detection requires the selected level to be stable for at least two times the internally gated interface clock period to trigger a synchronous interrupt.

As the module is synchronous, latency is minimal between the expected event occurrence and the activation of the interrupt line(s). This should not exceed 3 internally gated interface clock cycles + 2 interface clock cycles when the debouncing feature is not used. When the debouncing feature is active, the latency depends on the GPIO_DEBOUNCINGTIME register value and should be less than 3 internally gated interface clock cycles + 2 interface clock cycles + GPIO_DEBOUNCINGTIME value debouncing clock cycles + 3 debouncing clock cycles.

Figure 24-3. Interrupt Request Generation

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

24.3.3.3 Asynchronous Path: Wake-up Request Generation

In Idle mode (interface clock is shut down, the GPIO configuration registers have been previously programmed), an asynchronous path (Figure 24-4) detects the expected transition(s) on input GPIO (according to the registers programming) and activates an asynchronous wake-up request if the wakeup enable register is set. As shown in , there is only one external wake-up line, since all wake-up sources are merged together. Once the system is in wake up, the interface clock is re-started and, according to the input GPIO pin that triggered the wake-up request, the corresponding bits in the GPIO_IRQSTATUS_RAW_n registers are synchronously set to 1; on the following internal clock cycle, the interrupt lines 1 and/or 2 are active (active high) when the corresponding bits in GPIO_IRQSTATUS_SET_n registers are set.

NOTE: If debouncing is not enabled, there is no minimum input pulse width to trigger the wake-up request since there is no sampling operation.

If debouncing is used, the minimum pulse width is set by the debouncing specified time.

The ENAWAKEUP bit in the system configuration register (GPIO_SYSCONFIG) enables or disables the GPIO wake-up feature globally: if this bit is 0, the GPIO_IRQWAKEN has no effect.

Figure 24-4. Wake-Up Request Generation

Copyright © 2011, Texas Instruments Incorporated

24.3.3.4 Interrupt (or Wake-up) Line Release

When the host processor receives an interrupt request issued by the GPIO module, it can read the corresponding GPIO_IRQSTATUS_n register to find out which input GPIO has triggered the interrupt (or the wake-up request). After servicing the interrupt (or acknowledging the wake-up request), the processor resets the status bit and releases the interrupt line by writing a 1 in the corresponding bit of the GPIO_IRQSTATUS_n register. If there is still a pending interrupt request to serve (all bits in the GPIO_IRQSTATUS_RAW_n register not masked by the GPIO_IRQSTATUS_SET_n, which are not cleared by setting the GPIO_IRQSTATUS_CLR_n), the interrupt line will be re-asserted.

24.3.4 General-Purpose Interface Basic Programming Model

24.3.4.1 Power Saving by Grouping the Edge/Level Detection

Each GPIO module implements four gated clocks used by the edge/level detection logic to save power. Each group of eight input GPIO pins generates a separate enable signal depending on the edge/level detection register setting (because the input is 32 bits, four groups of eight inputs are defined for each GPIO module). If a group requires no edge/level detection, then the corresponding clock is gated (cut off). Grouping the edge/level enable can save the power consumption of the module as described in the following example.

If any of the registers:

- GPIO_LEVELDETECT0
- GPIO_LEVELDETECT1
- GPIO_RISINGDETECT
- GPIO_FALLINGDETECT

are set to 0101 0101h, then all clocks are active (power consumption is high);

are set to 0000 00FFh, then a single clock is active.

NOTE: When the clocks are enabled by writing to the GPIO_LEVELDETECT0, GPIO_LEVELDETECT1, GPIO_RISINGDETECT, and GPIO_FALLINGDETECT registers, the detection starts after 5 clock cycles. This period is required to clean the synchronization edge/level detection pipeline.

The mechanism is independent of each clock group. If the clock has been started before a new setting is performed, the following is recommended: first, set the new detection required; second, disable the previous setting (if necessary). In this way, the corresponding clock is not gated and the detection starts immediately.

24.3.4.2 Set and Clear Instructions

The GPIO module implements the set-and-clear protocol register update for the data output and interrupt enable, and wake-up enable registers. This protocol is an alternative to the atomic test and set operations and consists of writing operations at dedicated addresses (one address for setting bit[s] and one address for clearing bit[s]). The data to write is 1 at bit position(s) to clear (or to set) and 0 at unaffected bit(s).

Registers can be accessed in two ways:

- Standard: Full register read and write operations at the primary register address
- Set and clear (recommended): Separate addresses are provided to set (and clear) bits in registers. Writing 1 at these addresses sets (or clears) the corresponding bit into the equivalent register; writing a 0 has no effect.

Therefore, for these registers, three addresses are defined for one unique physical register. Reading these addresses has the same effect and returns the register value.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

24.3.4.2.1 Clear Instruction

24.3.4.2.1.1 Clear Interrupt Enable Registers (GPIO_IRQSTATUS_CLR_0 and GPIO_IRQSTATUS_CLR_1):

- A write operation in the clear interrupt enable1 (or enable2) register clears the corresponding bit in the interrupt enable1 (or enable2) register when the written bit is 1; a written bit at 0 has no effect.
- A read of the clear interrupt enable1 (or enable2) register returns the value of the interrupt enable1 (or enable2) register.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

24.3.4.2.1.2 Clear Wake-up Enable Register (GPIO_CLEARWKUENA):

- A write operation in the clear wake-up enable register clears the corresponding bit in the wake-up enable register when the written bit is 1; a written bit at 0 has no effect.
- A read of the clear wake-up enable register returns the value of the wake-up enable register.

24.3.4.2.1.3 Clear Data Output Register (GPIO_CLEARDATAOUT):

- A write operation in the clear data output register clears the corresponding bit in the data output register when the written bit is 1; a written bit at 0 has no effect.
- A read of the clear data output register returns the value of the data output register.

24.3.4.2.1.4 Clear Instruction Example

Assume the data output register (or one of the interrupt/wake-up enable registers) contains the binary value, 0000 0001 0000 0001h, and you want to clear bit 0.

With the clear instruction feature, write 0000 0000 0000 0001h at the address of the clear data output register (or at the address of the clear interrupt/wake-up enable register). After this write operation, a reading of the data output register (or the interrupt/wake-up enable register) returns 0000 0001 0000 0000h; bit 0 is cleared.

NOTE: Although the general-purpose interface registers are 32-bits wide, only the 16 least-significant bits are represented in this example.

Figure 24-5. Write @ GPIO_CLEARDATAOUT Register Example

24.3.4.2.2 Set Instruction

24.3.4.2.2.1 Set Interrupt Enable Registers (GPIO_IRQSTATUS_SET_0 and GPIO_IRQSTATUS_SET_1):

- A write operation in the set interrupt enable1 (or enable2) register sets the corresponding bit in the interrupt enable1 (or enable2) register when the written bit is 1; a written bit at 0 has no effect.
- A read of the set interrupt enable1 (or enable2) register returns the value of the interrupt enable1 (or enable2) register.

24.3.4.2.2.2 Set Wake-up Enable Register (GPIO_SETWKUENA):

- A write operation in the set wake-up enable register sets the corresponding bit in the wake-up enable register when the written bit is 1; a written bit at 0 has no effect.
- A read of the set wake-up enable register returns the value of the wake-up enable register.

24.3.4.2.2.3 Set Data Output Register (GPIO_SETDATAOUT):

- A write operation in the set data output register sets the corresponding bit in the data output register when the written bit is 1; a written bit at 0 has no effect.
- A read of the set data output register returns the value of the data output register.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

24.3.4.2.2.4 Set Instruction Example

Assume the interrupt enable1 (or enable2) register (or the data output register) contains the binary value, 0000 0001 0000 0000h, and you want to set bits 15, 3, 2, and 1.

With the set instruction feature, write 1000 0000 0000 1110h at the address of the set interrupt enable1 (or enable2) register (or at the address of the set data output register). After this write operation, a reading of the interrupt enable1 (or enable2) register (or the data output register) returns 1000 0001 0000 1110h; bits 15, 3, 2, and 1 are set.

NOTE: Although the general-purpose interface registers are 32-bits wide, only the 16 least-significant bits are represented in this example.

Figure 24-6. Write @ GPIO_SETIRQENABLEx Register Example

24.3.4.3 Data Input (Capture)/Output (Drive)

The output enable register (GPIO_OE) controls the output/input capability for each pin. At reset, all the GPIO-related pins are configured as input and output capabilities are disabled. This register is not used within the module; its only function is to carry the pads configuration.

When configured as an output (the desired bit reset in GPIO_OE), the value of the corresponding bit in the GPIO_DATAOUT register is driven on the corresponding GPIO pin. Data is written to the data output register synchronously with the interface clock. This register can be accessed with read/write operations or by using the alternate set and clear protocol register update feature. This feature lets you set or clear specific bits of this register with a single write access to the set data output register (GPIO_SETDATAOUT) or to the clear data output register (GPIO_CLEARDATAOUT) address. If the application uses a pin as an output and does not want interrupt/wake-up generation from this pin, the application must properly configure the wake-up enable and the interrupt enable registers.

When configured as an input (the desired bit set to 1 in GPIO_OE), the state of the input can be read from the corresponding bit in the GPIO_DATAIN register. The input data is sampled synchronously with the interface clock and then captured in the data input register synchronously with the interface clock. When the GPIO pin levels change, they are captured into this register after two interface clock cycles (the required cycles to synchronize and to write data). If the application uses a pin as an input, the application must properly configure the wake-up enable and the interrupt enable registers to the interrupt and wake-up feature as needed.

24.3.4.4 Debouncing Time

To enable the debounce feature for a pin, the GPIO configuration registers must be programmed as follows:

- The GPIO pin must be configured as input in the output enable register (write 1 to the corresponding bit of the GPIO_OE register).
- The debouncing time must be set in the debouncing value register (GPIO_DEBOUNCINGTIME). The GPIO_DEBOUNCINGTIME register is used to set the debouncing time for all input lines in the GPIO module. The value is global for all the ports of one GPIO module, so up to six different debouncing values are possible. The debounce cell is running with the debounce clock (32 kHz). This register represents the number of the clock cycle(s) (one cycle is 31 microseconds long) to be used.

  The following formula describes the required input stable time to be propagated to the debounced output:

Copyright © 2011, Texas Instruments Incorporated

Debouncing time = (DEBOUNCETIME + 1) × 31 µs

Where the DEBOUNCETIME field value in the GPIO_DEBOUNCINGTIME register is from 0 to 255.

- The debouncing feature must be enabled in the debouncing enable register (write 1 to the corresponding DEBOUNCEENABLE bit in the GPIO_DEBOUNCENABLE register).

24.3.4.5 GPIO as a Keyboard Interface

The general-purpose interface can be used as a keyboard interface (Figure 24-7). You can dedicate channels based on the keyboard matrix = * c). Figure 24-7 shows row channels configured as inputs with the input debounce feature enabled. The row channels are driven high with an external pull-up. Column channels are configured as outputs and drive a low level.

When a keyboard matrix key is pressed, the corresponding row and column lines are shorted together and a low level is driven on the corresponding row channel. This generates an interrupt based on the proper configuration (see Section 24.3.3).

When the keyboard interrupt is received, the processor can disable the keyboard interrupt and scan the column channels for the key coordinates.

- The scanning sequence has as many states as column channels: For each step in the sequence, the processor drives one column channel low and the others high.
- The processor reads the values of the row channels and thus detects which keys in the column are pressed.

At the end of the scanning sequence, the processor establishes which keys are pressed. The keyboard interface can then be reconfigured in the interrupt waiting state.

Figure 24-7. General-Purpose Interface Used as a Keyboard Interface

Copyright © 2011, Texas Instruments Incorporated

24.4 GPIO Registers

All module registers are 8-, 16-, or 32-bit accessible through the L4 interconnect (little-endian encoding). Access to registers is direct; no shadow registers are implemented.

Table 24-5. GPIO Registers

Address Offset	Acronym	Register Description	Section
0h	GPIO_REVISION	GPIO Revision Register	Section 24.4.1
10h	GPIO_SYSCONFIG	System Configuration Register	Section 24.4.2
20h	GPIO_EOI	End of Interrupt Register	Section 24.4.3
24h	GPIO_IRQSTATUS_RAW_0	Status Raw Register for Interrupt 1	Section 24.4.4
28h	GPIO_IRQSTATUS_RAW_1	Status Raw Register for Interrupt 2	Section 24.4.4
2Ch	GPIO_IRQSTATUS_0	Status Register for Interrupt 1	Section 24.4.5
30h	GPIO_IRQSTATUS_1	Status Register for Interrupt 2	Section 24.4.5
34h	GPIO_IRQSTATUS_SET_0	Enable Set Register for Interrupt 1	Section 24.4.6
38h	GPIO_IRQSTATUS_SET_1	Enable Set Register for Interrupt 2	Section 24.4.6
3Ch	GPIO_IRQSTATUS_CLR_0	Enable Clear Register for Interrupt 1	Section 24.4.7
40h	GPIO_IRQSTATUS_CLR_1	Enable Clear Register for Interrupt 2	Section 24.4.7
44h	GPIO_IRQWAKEN_0	Wakeup Enable Register for Interrupt 1	Section 24.4.8
48h	GPIO_IRQWAKEN_1	Wakeup Enable Register for Interrupt 2	Section 24.4.8
114h	GPIO_SYSSTATUS	System Status Register	Section 24.4.9
130h	GPIO_CTRL	Module Control Register	Section 24.4.10
134h	GPIO_OE	Output Enable Register	Section 24.4.11
138h	GPIO_DATAIN	Data Input Register	Section 24.4.12
13Ch	GPIO_DATAOUT	Data Output Register	Section 24.4.13
140h	GPIO_LEVELDETECT0	Low-level Detection Enable Register	Section 24.4.14
144h	GPIO_LEVELDETECT1	High-level Detection Enable Register	Section 24.4.15
148h	GPIO_RISINGDETECT	Rising-edge Detection Enable Register	Section 24.4.16
14Ch	GPIO_FALLINGDETECT	Falling-edge Detection Enable Register	Section 24.4.17
150h	GPIO_DEBOUNCENABLE	Debounce Enable Register	Section 24.4.18
154h	GPIO_DEBOUNCINGTIME	Debouncing Time Register	Section 24.4.19
190h	GPIO_CLEARDATAOUT	Clear Data Output Register	Section 24.4.20
194h	GPIO_SETDATAOUT	Set Data Output Register	Section 24.4.21

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

24.4.1 GPIO_REVISION Register

The GPIO revision register is a read only register containing the revision number of the GPIO module. A write to this register has no effect, the same as the reset. The GPIO_REVISION register is shown in Figure 24-8 and described in Table 24-6.

Figure 24-8. GPIO_REVISION Register

31–30	29–28	27–16
SCHEME	Reserved	FUNC
R-1	R-1	R-60h

15–11	10–8	7–6	5–0
RTL	MAJOR	CUSTOM	MINOR
R-1	R-0	R-0	R-1

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 24-6. GPIO_REVISION Register Field Descriptions

Bit	Field	Value	Description
31-30	SCHEME	1	Used to distinguish between old Scheme and current.
29-28	Reserved	1	Reserved
27-16	FUNC	0-FFFh	Indicates a software compatible module family.
15-11	RTL	0-1Fh	RTL version
10-8	MAJOR	0-7h	Major Revision
7-6	CUSTOM	0-3h	Indicates a special version for a particular device.
5-0	MINOR	0-3Fh	Minor Revision

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

24.4.2 GPIO_SYSCONFIG Register

The GPIO_SYSCONFIG register controls the various parameters of the L4 interconnect. The GPIO_SYSCONFIG register is shown in Figure 24-9 and described in Table 24-7.

NOTE: When the AUTOIDLE bit is set, the GPIO_DATAIN read command has a 3 OCP cycle latency due to the data in sample gating mechanism. When the AUTOIDLE bit is not set, the GPIO_DATAIN read command has a 2 OCP cycle latency.

Figure 24-9. GPIO_SYSCONFIG Register

31					16
Reserved					
R-0					

15	5	4	3	2	1	0
Reserved		IDLEMODE		ENAWAKEUPReserved	SOFTRESET	AUTOIDLE
R-0		R/W-0		R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 24-7. GPIO_SYSCONFIG Register Field Descriptions

Bit	Field	Value	Description
31-5	Reserved	0	Reserved
4-3	IDLEMODE		Power Management, Req/Ack control
		0	Force-idle. An idle request is acknowledged unconditionally
		1h	No-idle. An idle request is never acknowledged
		2h	Smart-idle. Acknowledgment to an idle request is given based on the internal activity of the module
		3h	Smart-idle Wakeup.
2	Reserved	0	Reserved. Always write the default value for future device compatibility.
2	ENAWAKEUP		Wake-up capability enabled/disabled
		0	Wake-up disable
		1	Wake-up enabled upon expected transition on input GPIO pin
1	SOFTRESET		Software reset. This bit is automatically reset by the hardware. During reads, it always returns 0.
		0	Normal mode
		1	The module is reset
0	AUTOIDLE		Internal interface clock gating strategy
		0	Internal Interface OCP clock is free-running
		1	Automatic internal OCP clock gating, based on the OCP interface activity

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

24.4.3 GPIO_EOI Register

The GPIO_EOI register provides software end of interrupt (EOI) control. The GPIO_EOI register is shown in Figure 24-10 and described in Table 24-8.

Figure 24-10. GPIO_EOI Register

31	16
Reserved	
R-0	

15	1	0
Reserved		LINE_NUMBER
R-0		R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 24-8. GPIO_EOI Register Field Descriptions

Bits	Field	Value	Description
31-1	Reserved	0	Reserved
0	LINE_NUMBER		Software End Of Interrupt (EOI) control. Write number of interrupt output.
		Write 0	EOI for interrupt output line #0
		Write 1	EOI for interrupt output line #1
		Read 0	Reads always 0 (no EOI memory)

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

24.4.4 GPIO_IRQSTATUS_RAW_n Register

The GPIO_IRQSTATUS_RAW_n register provides core status information for the interrupt handling, showing all active events (enabled and not enabled). The fields are read-write. Writing a 1 to a bit sets it to 1, that is, triggers the IRQ (mostly for debug). Writing a 0 has no effect, that is, the register value is not be modified. Only enabled, active events trigger an actual interrupt request on the IRQ output line. The GPIO_IRQSTATUS_RAW_n register is shown in Figure 24-11 and described in Table 24-9.

Figure 24-11. GPIO_IRQSTATUS_RAW_n Register

31	0
INTLINE[*n*]	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 24-9. GPIO_IRQSTATUS_RAW_n Register Field Descriptions

Bits	Field	Value	Description
31-0	INTLINE[*n*]		Interrupt n status.
		0	No effect
		1	IRQ is triggered.

24.4.5 GPIO_IRQSTATUS_n Register

The GPIO_IRQSTATUS_n register provides core status information for the interrupt handling, showing all active events (enabled and not enabled). The fields are read-write. Writing a 1 to a bit clears the bit to 0, that is, clears the IRQ. Writing a 0 has no effect, that is, the register value is not modified. Only enabled, active events trigger an actual interrupt request on the IRQ output line. The GPIO_IRQSTATUS_n register is shown in Figure 24-12 and described in Table 24-10.

Figure 24-12. GPIO_IRQSTATUS_n Register

31	0
INTLINE[*n*]	
R/W1C-0	

LEGEND: R/W = Read/Write; W1C = Write a 1 to clear to 0, a write of 0 has no effect; -*n* = value after reset

Table 24-10. GPIO_IRQSTATUS_n Register Field Descriptions

Bits	Field	Value	Description
31-0	INTLINE[*n*]		Interrupt n status.
		0	No effect
		1	IRQ is triggered.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

24.4.6 GPIO_IRQSTATUS_SET_n Register

All 1-bit fields in the GPIO_IRQSTATUS_SET_n register enable a specific interrupt event to trigger an interrupt request. Writing a 1 to a bit enables the interrupt field. Writing a 0 has no effect, that is, the register value is not modified. The GPIO_IRQSTATUS_SET_n register is shown in Figure 24-13 and described in Table 24-11.

Figure 24-13. GPIO_IRQSTATUS_SET_n Register

31	0
INTLINE[*n*]	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 24-11. GPIO_IRQSTATUS_SET_n Register Field Descriptions

Bits	Field	Value	Description
31-0	INTLINE[*n*]		Interrupt n enable
		0	No effect
		1	Enable IRQ generation.

24.4.7 GPIO_IRQSTATUS_CLR_n Register

All 1-bit fields in the GPIO_IRQSTATUS_CLR_n register clear a specific interrupt event. Writing a 1 to a bit disables the interrupt field. Writing a 0 has no effect, that is, the register value is not modified. The GPIO_IRQSTATUS_CLR_n register is shown in Figure 24-14 and described in Table 24-12.

Figure 24-14. GPIO_IRQSTATUS_CLR_n Register

31	0
INTLINE[*n*]	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 24-12. GPIO_IRQSTATUS_CLR_n Register Field Descriptions

Bits	Field	Value	Description
31-0	INTLINE[*n*]		Interrupt n enable
		0	No effect
		1	Disable IRQ generation.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

24.4.8 GPIO_IRQWAKEN_n Register

All 1-bit fields in the GPIO_IRQWAKEN_n register enable a specific (synchronous) IRQ request source to generate an asynchronous wakeup (on the appropriate wakeup line). The GPIO_IRQWAKEN_n register allows you to mask the expected transition on input GPIO from generating a wakeup request. The GPIO_IRQWAKEN_n register is programmed synchronously with the interface clock before any Idle mode request coming from the host processor. The GPIO_IRQWAKEN_n register is shown in Figure 24-15 and described in Table 24-13.

NOTE: In Force-Idle mode, the module wake-up feature is totally inhibited. The wake-up generation can also be gated at module level using the EnaWakeUp bit from the GPIO_SYSCONFIG register.

Figure 24-15. GPIO_IRQWAKEN_n Register

31	0
INTLINE[*n*]	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 24-13. GPIO_IRQWAKEN_n Register Field Descriptions

Bits	Field	Value	Description
31-0	INTLINE[*n*]		Wake Up Enable for Interrupt Line
		0	No effect
		1	Enable wake-up generation.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

24.4.9 GPIO_SYSSTATUS Register

The GPIO_SYSSTATUS register provides the reset status information about the GPIO module. It is a read-only register; a write to this register has no effect. The GPIO_SYSSTATUS register is shown in Figure 24-16 and described in Table 24-14.

Figure 24-16. GPIO_SYSSTATUS Register

31		16
Reserved		
R-0		

15	1	0
Reserved		RESETDONE
R-0		R-0

LEGEND: R = Read only; -*n* = value after reset

Table 24-14. GPIO_SYSSTATUS Register Field Descriptions

Bit	Field	Value	Description
31-1	Reserved	0	Reserved
0	RESETDONE		Reset status information.
		0	Internal Reset is on-going
		1	Reset completed

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

24.4.10 GPIO_CTRL Register

The GPIO_CTRL register controls the clock gating functionality. The DISABLEMODULE bit controls a clock gating feature at the module level. When set, this bit forces the clock gating for all internal clock paths. Module internal activity is suspended. System interface is not affected by this bit. System interface clock gating is controlled with the AUTOIDLE bit in the system configuration register (GPIO_SYSCONFIG). This bit is to be used for power saving when the module is not used because of the multiplexing configuration selected at the chip level. This bit has precedence over all other internal configuration bits. The GPIO_CTRL register is shown in Figure 24-17 and described in Table 24-15.

Figure 24-17. GPIO_CTRL Register

31–16	15–3	2–1	0
Reserved	Reserved	GATINGRATIO	DISABLEMODULE
R-0	R-0	R/W-1	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 24-15. GPIO_CTRL Register Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Reserved
2-1	GATINGRATIO		Gating Ratio. Controls the clock gating for the event detection logic.
		0	Functional clock is interface clock.
		1h	Functional clock is interface clock divided by 2.
		2h	Functional clock is interface clock divided by 4.
		3h	Functional clock is interface clock divided by 8.
0	DISABLEMODULE		Module Disable
		0	Module is enabled, clocks are not gated.
		1	Module is disabled, clocks are gated.

24.4.11 GPIO_OE Register

The GPIO_OE register is used to enable the pins output capabilities. At reset, all the GPIO related pins are configured as input and output capabilities are disabled. This register is not used within the module, its only function is to carry the pads configuration. When the application is using a pin as an output and does not want interrupt/wake-up generation from this pin, the application can/has to configure properly the Wakeup Enable and the Interrupt Enable registers. The GPIO_OE register is shown in Figure 24-18 and described in Table 24-16.

Figure 24-18. GPIO_OE Register

31–0
OUTPUTEN[*n*]
R/W-FFFF FFFFh

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 24-16. GPIO_OE Register Field Descriptions

Bits	Field	Value	Description
31-0	OUTPUTEN[*n*]		Output Data Enable
		0	The corresponding GPIO port is configured as an output.
		1	The corresponding GPIO port is configured as an input.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

24.4.12 GPIO_DATAIN Register

The GPIO_DATAIN register is used to register the data that is read from the GPIO pins. The GPIO_DATAIN register is a read-only register. The input data is sampled synchronously with the interface clock and then captured in the GPIO_DATAIN register synchronously with the interface clock. So, after changing, GPIO pin levels are captured into this register after two interface clock cycles (the required cycles to synchronize and to write the data). The GPIO_DATAIN register is shown in Figure 24-19 and described in Table 24-17.

NOTE: When the AUTOIDLE bit in the system configuration register (GPIO_SYSCONFIG) is set, the GPIO_DATAIN read command has a 3 OCP cycle latency due to the data in sample gating mechanism. When the AUTOIDLE bit is not set, the GPIO_DATAIN read command has a 2 OCP cycle latency.

Figure 24-19. GPIO_DATAIN Register

31	0
DATAIN	
R-0	

LEGEND: R = Read only; -*n* = value after reset

Table 24-17. GPIO_DATAIN Register Field Descriptions

Bits	Field	Value	Description
31-0	DATAIN	0-FFFF FFFFh	Sampled Input Data

24.4.13 GPIO_DATAOUT Register

The GPIO_DATAOUT register is used for setting the value of the GPIO output pins. Data is written to the GPIO_DATAOUT register synchronously with the interface clock. This register can be accessed with direct read/write operations or using the alternate 'Set/Clear' feature. This feature enables to set or clear specific bits of this register with a single write access to the set data output register (GPIO_SETDATAOUT) or to the clear data output register (GPIO_CLEARDATAOUT) address. The GPIO_DATAOUT register is shown in Figure 24-20 and described in Table 24-18.

Figure 24-20. GPIO_DATAOUT Register

31	0
DATAOUT	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 24-18. GPIO_DATAOUT Register Field Descriptions

Bits	Field	Value	Description
31-0	DATAOUT	0-FFFF FFFFh	Data to set on output pins

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

24.4.14 GPIO_LEVELDETECT0 Register

The GPIO_LEVELDETECT0 register is used to enable/disable for each input lines the low-level (0) detection to be used for the interrupt request generation. The GPIO_LEVELDETECT0 register is shown in Figure 24-21 and described in Table 24-19.

NOTE: Enabling at the same time high-level detection and low-level detection for one given pin makes a constant interrupt generator.

Figure 24-21. GPIO_LEVELDETECT0 Register

31	0
LEVELDETECT0[*n*]	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 24-19. GPIO_LEVELDETECT0 Register Field Descriptions

Bits	Field	Value	Description
31-0	LEVELDETECT0[*n*]		Low Level Interrupt Enable
		0	Disable the IRQ assertion on low-level detect.
		1	Enable the IRQ assertion on low-level detect.

24.4.15 GPIO_LEVELDETECT1 Register

The GPIO_LEVELDETECT1 register is used to enable/disable for each input lines the high-level (1) detection to be used for the interrupt request generation. The GPIO_LEVELDETECT1 register is shown in Figure 24-22 and described in Table 24-20.

NOTE: Enabling at the same time high-level detection and low-level detection for one given pin makes a constant interrupt generator.

Figure 24-22. GPIO_LEVELDETECT1 Register

31	0
LEVELDETECT1[*n*]	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 24-20. GPIO_LEVELDETECT1 Register Field Descriptions

Bits	Field	Value	Description
31-0	LEVELDETECT1[*n*]		High Level Interrupt Enable
		0	Disable the IRQ assertion on high-level detect.
		1	Enable the IRQ assertion on high-level detect.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

24.4.16 GPIO_RISINGDETECT Register

The GPIO_RISINGDETECT register is used to enable/disable for each input lines the rising-edge (transition 0 to 1) detection to be used for the interrupt request generation. The GPIO_RISINGDETECT register is shown in Figure 24-23 and described in Table 24-21.

Figure 24-23. GPIO_RISINGDETECT Register

31	0
RISINGDETECT[*n*]	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 24-21. GPIO_RISINGDETECT Register Field Descriptions

Bits	Field	Value	Description
31-0	RISINGDETECT[*n*]		Rising Edge Interrupt Enable
		0	Disable IRQ on rising-edge detect.
		1	Enable IRQ on rising-edge detect.

24.4.17 GPIO_FALLINGDETECT Register

The GPIO_FALLINGDETECT register is used to enable/disable for each input lines the falling-edge (transition 1 to 0) detection to be used for the interrupt request generation. The GPIO_FALLINGDETECT register is shown in Figure 24-24 and described in Table 24-22.

Figure 24-24. GPIO_FALLINGDETECT Register

31	0
FALLINGDETECT[*n*]	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 24-22. GPIO_FALLINGDETECT Register Field Descriptions

Bits	Field	Value	Description
31-0	FALLINGDETECT[*n*]		Falling Edge Interrupt Enable
		0	Disable IRQ on falling-edge detect.
		1	Enable IRQ on falling-edge detect.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

24.4.18 GPIO_DEBOUNCENABLE Register

The GPIO_DEBOUNCENABLE register is used to enable/disable the debouncing feature for each input line. The GPIO_DEBOUNCENABLE register is shown in Figure 24-25 and described in Table 24-23.

Figure 24-25. GPIO_DEBOUNCENABLE Register

31	0
DEBOUNCEENABLE[*n*]	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 24-23. GPIO_DEBOUNCENABLE Register Field Descriptions

Bits	Field	Value	Description
31-0	DEBOUNCEENABLE[*n*]		Input Debounce Enable
		0	Disable debouncing feature on the corresponding input port.
		1	Enable debouncing feature on the corresponding input port.

24.4.19 GPIO_DEBOUNCINGTIME Register

The GPIO_DEBOUNCINGTIME register controls debouncing time (the value is global for all ports). The debouncing cell is running with the debouncing clock (32 kHz), this register represents the number of the clock cycle(s) (31µs long) to be used. The GPIO_DEBOUNCINGTIME register is shown in Figure 24-26 and described in Table 24-24.

Figure 24-26. GPIO_DEBOUNCINGTIME Register

31	16
Reserved	
R-0	

15	8	7	0
Reserved		DEBOUNCETIME	
R-0		R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 24-24. GPIO_DEBOUNCINGTIME Register Field Descriptions

Bits	Field	Value	Description
31-8	Reserved	0	Reserved
7-0	DEBOUNCETIME	0-FFh	Input Debouncing Value in 31 microsecond steps. Debouncing time = (DEBOUNCETIME + 1) × 31 microseconds

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

24.4.20 GPIO_CLEARDATAOUT Register

Writing a 1 to a bit in the GPIO_CLEARDATAOUT register clears to 0 the corresponding bit in the GPIO_DATAOUT register; writing a 0 has no effect. The GPIO_CLEARDATAOUT register is shown in Figure 24-27 and described in Table 24-25.

Figure 24-27. GPIO_CLEARDATAOUT Register

31	0
INTLINE [*n*]	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 24-25. GPIO_CLEARDATAOUT Register Field Descriptions

Bits	Field	Value	Description
31-0	INTLINE [*n*]		Clear Data Output Register
		0	No effect
		1	Clear the corresponding bit in the GPIO_DATAOUT register.

24.4.21 GPIO_SETDATAOUT Register

Writing a 1 to a bit in the GPIO_SETDATAOUT register sets to 1 the corresponding bit in the GPIO_DATAOUT register; writing a 0 has no effect. A read of the GPIO_SETDATAOUT register returns the value of the data output register (GPIO_DATAOUT). The GPIO_SETDATAOUT register is shown in Figure 24-28 and described in Table 24-26.

Figure 24-28. GPIO_SETDATAOUT Register

31	0
INTLINE [*n*]	
R/W-0	

LEGEND: R/W = Read/Write; -*n* = value after reset

Table 24-26. GPIO_SETDATAOUT Register Field Descriptions

Bits	Field	Value	Description
31-0	INTLINE [*n*]		Set Data Output Register
		0	No effect
		1	Set the corresponding bit in the GPIO_DATAOUT register.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Chapter 25

SPRUH73–July 2011

Initialization

This chapter describes the initialization of the device.

Topic **Page**

25.1 Introduction **2946**
25.2 Integration **2946**
25.3 Functional Description **2946**
25.4 Use Cases **2982**
25.5 Registers **2982**

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

25.1 Introduction

25.2 Integration

25.3 Functional Description

This document describes the booting functionality of the device, referred hereafter as ROM Code. The booting functionality covers the following features:

Memory Booting: Booting the device by starting code stored on permanent memories like flash-memory or memory cards. This process is usually performed after either device cold or warm reset.

Peripheral Booting: Booting the device by downloading the executable code over a communication interface like UART, USB or Ethernet. This process is intended for flashing a device.

The device always starts up in secure mode. The ROM Code takes care of early initialization. The ROM code switches the device into public mode. Hence the Public ROM Code provides run-time services for cache maintenance.

The document doesn't cover in details the security related aspects. This is covered separately in the Secure ROM specification. It is intended for all who design systems based on this device and develop boot loader images and flashing tools.

25.3.1 Device Types

This device has two types of production devices: a High Secure (**HS**) device and a General Purpose (**GP**) device. The fundamental difference between these two types of production devices is the approach to secure booting. The **HS Device** doesn't allow booting code which is not trusted (authenticated, that is digitally signed and checked for integrity). On the other hand the **GP Device** has its security features disabled and does not require signed code for booting the device. Image formats are different depending on the device type and described in this document.

25.3.2 Architecture

The architecture of the Public ROM Code is shown in Figure 25-1. It is split into three main layers with a top-down approach: high-level, drivers, and hardware abstraction layer (HAL). One layer communicates with a lower level layer through a unified interface.

The high level layer is in charge of the main tasks of the Public ROM Code: watchdog & clocks configuration and main booting routine.

The drivers layer implements the logical and communication protocols for any booting device in accordance with the interface specification.

Finally the HAL implements the lowest level code for interacting with the hardware infrastructure IPs. End booting devices are attached to device IO pads.

Figure 25-1. Public ROM Code Architecture

25.3.3 Functionality

Figure 25-2 illustrates the high level flow for the Public ROM Code booting procedure. On this device the Public ROM Code starts upon completion of the secure startup (performed by the Secure ROM Code). The ROM Code then performs platform configuration and initialization as part of the public start-up procedure.

The booting device list is created based on the **SYSBOOT** pins. A booting device can be a **memory booting device** (soldered flash memory or temporarily booting device like memory card) or a **peripheral interface** connected to a host.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

The main loop of the booting procedure goes through the booting device list and tries to search for an image from the currently selected booting device. This loop is exited if a valid booting image is found and successfully executed or upon watchdog expiration.

The image authentication procedure is performed prior to image execution on an **HS Device**. Failure in authentication procedure leads to branching to a “dead loop” in Secure ROM (waiting for a watchdog reset).

Figure 25-2. Public ROM Code Boot Procedure

25.3.4 Memory Map

25.3.4.1 Public ROM Memory Map

The on-chip ROM memory map is shown in Figure 25-3. The bottom part of the physical ROM contains the Secure ROM Code. The upper part holds the Public ROM Code. The Public ROM Code mapping consists in the following:

- Exception vectors
- CRC
- Dead loops collection
- Code and const data sections
- ROM Version

Figure 25-3. ROM Memory Map

Public ROM Exception Vectors

Table 25-1 lists the Public ROM exception vectors. The reset exception is redirected to the Public ROM Code startup. Other exceptions are redirected to their RAM handlers by loading appropriate addresses into the PC register.

Table 25-1. ROM Exception Vectors

Address	Exception	Content
20000h	Reset	Branch to the Public ROM Code startup
20004h	Undefined	PC = 4030D004h
20008h	SWI	PC = 4030D008h
2000Ch	Pre-fetch abort	PC = 4030D00Ch
20010h	Data abort	PC = 4030D010h
20014h	Unused	PC = 4030D014h
20018h	IRQ	PC = 4030D018h
2001Ch	FIQ	PC = 4030D01Ch

Public ROM Code CRC

The Public ROM Code CRC is calculated as 32 bit CRC code (CRC-32-IEEE 802.3) for the address range 20000h – 2BFFFh. The four bytes CRC code is stored at location 20020h.

Dead Loops

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Built-in dead loops are used for different purposes as shown in below Table 25-2. All dead loops are branch instructions coded in ARM mode. The fixed location of these dead loops facilitates debugging and testing. The first seven dead loops are default exception handlers linked with RAM exception vectors. The dead loops might be called directly from the user code. However there exists a special function which can be called from ROM Code in order to execute a dead loop. The function is an assembly code in ARM mode which takes the dead loop address from R0 register. The main purpose of the function is to go a dead loop, till the watchdog expires and resets the device, when device is **GP** type. In case of **HS Device** the function branches to the requested dead loop. The function is located at address 200C0h. In addition the function clears global cold reset status upon issuing the global SW reset.

Table 25-2. Dead Loops

Address	Purpose
20080h	Undefined exception default handler
20084h	SWI exception default handler
20088h	Pre-fetch abort exception default handler
2008Ch	Data abort exception default handler
20090h	Unused exception default handler
20094h	IRQ exception default handler
20098h	FIQ exception default handler
2009Ch	Validation tests PASS
200A0h	Validation tests FAIL
200A4h	Reserved
200A8h	Image not executed or returned.
200ACh	Reserved
200B0h	Reserved
200B4h	Reserved
200B8h	Reserved
200BCh	Reserved

Code

This space is used to hold code and constant data.

Public ROM Code Version

The ROM Code version consists of two decimal numbers: major and minor. It can be used to identify the ROM Code release version in a given IC (e.g., useful to identify an IC version). The ROM Code version is a 32bits hexadecimal value located at address 2BFFCh.

25.3.4.2 Public RAM Memory Map

The Public ROM Code makes use of the on chip RAM module connected to the L3 interconnect (further referred as **L3 RAM** in this document). Its usage is shown in Figure 25-4. The Public RAM memory map ranges from address 402F0000h to 4030FFFFh on a **GP Device**.

Figure 25-4. Public RAM Memory Map

Download Image

This area is used by the Public ROM Code to store the downloaded boot image. It can be up to 110KB on **GP Device** and 46KB for **HS device**.

Public Stack

Space reserved for stack.

RAM Exception Vectors

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

The RAM exception vectors enable a simple means for redirecting exceptions to custom handlers. Table 25-3 shows content of the RAM space reserved for RAM vectors. The first seven addresses are ARM instructions which load the value located in the subsequent seven addresses into the PC register. Theses instructions are executed when an exception occurs since they are called from the ROM exception vectors. Undefined, SWI, Unused and FIQ exceptions are redirected to a hardcoded dead loop. Pre-fetch abort, data abort, and IRQ exception are redirected to pre-defined ROM handlers. User code can redirect any exception to a custom handler either by writing its address to the appropriate location from 4030D024h to 4030D03Ch or by overriding the branch (load into PC) instruction between addresses from 4030D004h to 4030D01Ch.

Table 25-3. RAM Exception Vectors

Address	Exception	Content
4030D000h	Reserved	Reserved
4030D004h	Undefined	PC = [4030D024h]
4030D008h	SWI	PC = [4030D028h]
4030D00Ch	Pre-fetch abort	PC = [4030D02Ch]
4030D010h	Data abort	PC = [4030D030h]
4030D014h	Unused	PC = [4030D034h]
4030D018h	IRQ	PC = [4030D038h]
4030D01Ch	FIQ	PC = [4030D03Ch]
4030D020h	Reserved	20090h
4030D024h	Undefined	20080h
4030D028h	SWI	20084h
4030D02Ch	Pre-fetch abort	Address of default pre-fetch abort handler[(1)]
4030D030h	Data abort	Address of default data abort handler[(1)]
4030D034h	Unused	20090h
4030D038h	IRQ	Address of default IRQ handler
4030D03Ch	FIQ	20098h

(1) The default handlers for pre-fetch and data abort are performing reads from CP15 debug registers to retrieve the reason of the abort:
- In case of pre-fetch abort: the IFAR register is read from CP15 and stored into R0. The IFSR register is read and stored into the R1 register. Then the ROM Code jumps to the pre-fetch abort dead loop (20088h).
- In case of data abort: the DFAR register is read from CP15 and stored into R0. The DFSR register is read and stored into the R1 register. Then the ROM Code jumps to the data abort dead loop (2008Ch).

Tracing Data

This area contains trace vectors reflecting the execution path of the public boot. Section 25.3.13 describes the usage of the different trace vectors and lists all the possible trace codes.

Table 25-4. Tracing Data

Address	Size[bytes]	Description
4030D040h	4	Current tracing vector, word 1
4030D044h	4	Current tracing vector, word 2
4030D048h	4	Current tracing vector, word 3
4030D04Ch	4	Current copy of the PRM_RSTST register (reset reasons)
4030D050h	4	Cold reset run tracing vector, word 1
4030D054h	4	Cold reset run tracing vector, word 2
4030D058h	4	Cold reset run tracing vector, word 3
4030D05Ch	4	Reserved
4030D060h	4	Reserved
4030D064h	4	Reserved

Copyright © 2011, Texas Instruments Incorporated

Static Variables

This area contains the ROM Code static variables used during boot time.

25.3.5 Start-up and Configuration

25.3.5.1 ROM Code Start-up

On this device the main MPU subsystem always starts its execution in secure mode after reset due to the **TrustZone architecture** (the Secure ROM Code implements the reset handler).

The Public ROM Code is physically located at the address 20000h that is immediately next to the Secure ROM Code.

Figure 25-5. ROM Code Startup Sequence

As shown at top of Figure 25-5, the CPU jumps to the Public ROM Code reset vector once it has completed the **secure boot** initialization.

Once in public mode the CPU performs the public-side initialization and stack setup (compiler auto generated C- initialization or “scatter loading”). Then it configures the watchdog timer 1 (set to three minutes), performs system clocks configuration. Finally it jumps to the booting routine.

25.3.5.2 CPU State at Public Startup

The CPU **L1 instruction cache** and **branch prediction** mechanisms are not activated as part of the public boot process. The public vector base address is configured to the reset vector of Public ROM Code (20000h). MMU is left switched off during the public boot (hence L1 data cache off).

25.3.5.3 Clocking Configuration

The device supports the following frequencies based on SYSBOOT[15:14]

Table 25-5. Crystal Frequences Supported

SYSBOOT[15:14]	Crystal Frequency
00b	19.2 MHz
01b	24 MHz
10b	25 MHz
11b	26 MHz

The ROM Code configures the clocks and DPLLs which are necessary for ROM Code execution:

- **L3 ADPLLS** locked to provide 200MHz clocks for peripheral blocks.
- **DDR DPLL** locked to provide 400MHz.
- **MPU ADPLLS** is locked to provide 500 MHz for the A8.
- **PER ADPLLJ** is locked to provide 960MHz and 192MHz for peripheral blocks.

The Table 16 summarizes the **ROM Code default settings** for clocks. This default configuration enables all the ROM Code functionalities with minimized needs on power during boot.

Table 16: The ROM Code default clock settings

Table 25-6.

Clock	Frequency [MHz]	Source
L3_FCLK	200	CORE_CLKOUTM4
SPI_CLK	48	PER_CLKOUTM2
MMC_CLK	96	PER_CLKOUTM2
UART_CLK	48	PER_CLKOUTM2

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Table 25-6. (continued)

I2C_CLK	48	PER_CLKOUTM2
MPU_CLK	500	MPU_PLL
USB_PHY_CLK	960MHz	PER_CLKDCOLDO

The DPLLs and PRCM clock dividers are configured with the ROM Code default values after cold or warm reset in order to give the same working conditions to the Public ROM Code sequence.

25.3.6 Booting

25.3.6.1 Overview

Figure 25-6 shows the booting procedure. First a **booting device list** is created. The list consists of all devices which will be searched for a booting image. The list is filled in based on the **SYSBOOT**.

Figure 25-6. ROM Code Booting Procedure

Once the booting device list is setup, the booting routine examines the devices enumerated in the list sequentially and either executes the **memory booting** or **peripheral booting** procedure depending on the **booting device type**. The memory booting procedure is executed when the booting device type is one of NOR, NAND, MMC or SPI-EEPROM. The peripheral booting is executed when the booting device type is Ethernet, USB or UART.

The memory booting procedure reads data from a memory type device. If a valid **booting image** is found and successfully read from the external memory device:

- If the device is an **HS device** then the image is **authenticated** and Initial SW **started** upon successful authentication.
- If the device is a **GP device** the Initial SW is simply **started** (no authentication involved).

The peripheral booting procedure downloads data from a host (commonly a PC) to the device device by means of Ethernet, USB or UART links. The ROM Code uses a host-slave logical protocol for synchronization. Upon successful UART, USB or Ethernet connection the host sends the image binary contents. The peripheral booting procedure is described in detail in Section 25.3.9.

If the memory or peripheral booting fails for all devices enumerated in the device list then the ROM Code gets into a loop, waiting for the watchdog to reset the system.

25.3.6.2 Device List

The ROM Code creates the **device list** based on information gathered from the **SYSBOOT configuration pins** sensed in the control module. The pins are used to index the device table from which the list of devices is extracted

25.3.6.2.1 SYSBOOT Configuration Pins

Table 16: SYSBOOT configuration pins

Table 25-7.

Boot Modes				SYSBOOT[4:0]
1st	2nd	3rd	4th	
reserved	reserved	**reserved**	**reserved**	00000
UART	**XIP w/ WAIT (MUX2)**	**MMC**	**SPI**	00001
UART	SPI	**NAND**	**NANDI2C**	00010
UART	SPI	**XIP (MUX2)**	MMC	00011
UART	**XIP w/ WAIT (MUX1)**	**MMC**	**NAND**	00100
UART	**XIP (MUX1)**	**SPI**	**NANDI2C**	00101

Copyright © 2011, Texas Instruments Incorporated

Table 25-7. (continued)

EMAC	SPI	**NAND**	**NANDI2C**	00110
EMAC	MMC	**XIP WAIT (MUX2)**	NAND	00111
EMAC	MMC	**XIP(MUX2)**	**NANDI2C**	01000
EMAC	**XIP WAIT (MUX1)**	**NAND**	**MMC**	01001
EMAC	**XIP (MUX1)**	**SPI**	**NANDI2C**	01010
USB	NAND	SPI	MMC	01011
USB	NAND	**XIP(MUX2)**	NANDI2C	01100
USB	NAND	**XIP(MUX1)**	SPI	01101
reserved	reserved	reserved	reserved	01110
GP Fast External Boot	UART	EMAC	reserved	01111
XIP (MUX1)	**UART**	**EMAC**	**MMC**	10000
XIP w/WAIT (MUX1)	**UART**	**EMAC**	**MMC**	10001
NAND	NANDI2C	USB	**UART**	10010
NAND	NANDI2C	**MMC**	**UART**	10011
NAND	NAND12C	**SPI**	**EMAC**	10100
NANDI2C	MMC	**EMAC**	**UART**	10101
SPI	MMC	UART	EMAC	10110
MMC	SPI	**UART**	**USB**	10111
SPI	MMC	USB	**UART**	11000
SPI	MMC	EMAC	**UART**	11001
XIP (MUX2)	UART	SPI	MMC	11010
XIP w/ WAIT (MUX2)	UART	SPI	MMC	11011
MMC1	MMC	**UART**	**USB**	11100
Reserved	**reserved**	**reserved**	**reserved**	11101
Reserved	reserved	**reserved**	**reserved**	11110
GP Fast external Boot	EMAC	**UART**	**reserved**	11111

The ROM Code uses the row pointed by the SYSBOOT pins value .The device list is filled in with the 1st to 4th devices.

Table 16 is the decoding table for SYSBOOT pin configuration. The following shortcuts are used in the table:

- MMC1– MMC or SD card (MMC port 1)
- MMC– MMC or SD card (MMC port 0)
- NAND / NANDI2C– NAND flash memory / read geometry from EEPROM on I2C0
- XIP / XIP w/ WAIT– NOR or other XIP device with or without wait monitoring
- UART– UART interface (UART port 0)
- EMAC– Ethernet interface (EMAC port 1)
- SPI– SPI EEPROM (SPI 0, CS0)
- USB– USB interface (USB0)

Note: For any SYSBOOT value that is selected, please be aware that the pin muxing implications. For eg, if the boot mode selected is EMAC, NAND, SPI, NANDI2C, the SOC will drive EMAC, GPMC, SPI and I2C pins, in that order, depending on which boot device finally succeeds. For specific details of the pins driven by each device, please refer the description of that boot device later in this document.

To extend the boot flow to boot from devices that are not natively supported by the ROM, SPI boot can be used. For example, if a customer wants to boot from a USB stick, the system can be configured to boot from a SPI flash, and the code for configuring the USB and booting from a USB stick can be loaded into the SPI flash. This is known as a secondary boot.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

The values mentioned in the last row and corresponding SYSBOOT[4:0]= x1111 when used on an EMULATOR device correspond to the HW assisted bypass mode booting feature offered on the device. In that case the ROM Code is fully bypassed and the device starts up from reset vector 0x0 routed to the GPMC. This feature is fully driven by HW without any SW involvement. In this configuration the SYSBOOT[9] signal selects whether the WAIT signal must be monitored on the GPMC interface. The external boot feature is not available on GP / HS production devices.

However in the case of **GP Device** a “Fast XIP boot mechanism” (as described in section 6) is provided where minimal execution is performed from ROM Code for configuring the GPMC interface and then directly jump to the code contained in the connected NOR flash device.

The NOR device must fulfill the following requirements for EMU external boot mode and GP fast XIP mode:

- Addr/Data muxed device or a non-muxed (selected using SYSBOOT[11:10]) device in connected in MUX1 configuration
- Bus width selected by SYSBOOT[8]
- CS0 chip select
- Device wait signal connected to the WAIT0 GPMC signal (if used)
- The wait monitoring enable/disable is based on the value of SYSBOOT[9]

25.3.7 Fast External Booting

25.3.7.1 Overview

The Fast External boot feature:

-
  - Is only possible on **GP Device**
  - Consists of a blind jump in ARM mode to a code located in an external XIP device connected to CS0
  - The jump is performed with minimum on-chip ROM Code execution, without configuring any PLL
  - Allows the customer to create its own booting code
  - Is set up by means of the configuration pins, see Table 16.

25.3.7.2 External Booting

Figure 25-7 shows the Fast External Boot procedure. The code does not make use of RAM and is designed for fast execution.

Figure 25-7. Fast External Boot

25.3.8 Memory Booting

25.3.8.1 Overview

The Memory Booting procedure takes care of starting an external code located in memory device types. These devices are also called **permanent booting devices** since the device uses them for booting permanently.

Figure 25-8. Memory Booting

The permanent booting devices supported are:

- MMC/SD cards
- NOR flash
- NAND flash
- SPI EEPROMs

Copyright © 2011, Texas Instruments Incorporated

There are two groups of permanent booting devices distinguished by the need of **code shadowing**. The code shadowing means copying a code from a non-directly addressable device into a location (typically a RAM area) from where the code can be executed. Devices which are directly addressable are called **eXecute In Place** (XIP) devices.

The Memory Booting flowchart is shown in Figure 25-8. The second step is about performing the shadowing of the image that is copying the image from external mass storage (non-XIP) into internal RAM. On an **HS Device**, image **authentication** and **execution** are the last steps performed. Unsuccessful authentication or return from image results in a “dead loop”. Failure in image copy results in Memory Booting returning to the main booting procedure which will select the next device for booting. The next sections detail procedures for device initialization & detection in addition to the description of the sector read routine for each supported device type. A sector is a logical unit of 512 bytes.

The detection of whether an image is present or not on a selected device is different depending on the device type:

- On an **HS Device** type a booting image is considered present when the image TOC structure begins with a known Initial SW TOC item.
- On a **GP Device** type a booting image is considered to be present when the first four bytes of the sector is not equal to 0000 0000h or FFFF FFFFh.

During the first read sector call, sectors are copied to a temporary RAM buffer. Once the image is found and destination address is known, the content of the temporary buffer is moved to the target RAM location so it is needed to re-read the first image sector. On a **GP Device** the GP header is discarded, therefore only executable code is located in RAM with the first executable instruction located at the destination address.

The Image authentication and execution is detailed in Section 25.3.11. For more information about image formats and contents, see Section 25.3.10

MMC/SD cards and NAND devices can hold up to four copies of the booting image. Therefore the ROM Code searches for one valid image out of the four if present by walking over the first four blocks of the mass storage space. Other XIP devices (NOR) use only one copy of the booting image.

25.3.8.2 XIP Memory

The ROM Code can boot directly from XIP devices. A typical XIP device is a NOR flash memory. Support for XIP devices is performed under the following assumptions:

- Uses GPMC as the communication interface
- Up to 1Gbit (128Mbytes) memories can be connected
- Both x8 and x16 data bus width
- Asynchronous protocol
- Supports address / data multiplexed mode and non-muxed mode
- GPMC clock is 50MHz
- Device connected to CS0 mapped to address 8000000h
- Wait pin signal WAIT0 monitored depending on the SYSBOOT pin configuration (XIP/XIPWAIT).
- Flexible muxing options for gpmc_a0-gpmc_a11 for non-muxed XIP devices

Depending on the SYSBOOT pins, the GPMC is configured to use the WAIT signal connected on the WAIT0 pin or not. Wait pin polarity is set to stall accessing memory when the WAIT0 pin is low. The wait monitoring is intended to be used with memories which require long time for initialization after reset or need to pause while reading data. The boot procedure from XIP device can be described as such:

- Configure GPMC for XIP device access.
- Set the image location to 8000000h
- Verify if bootable image is present at the image location.
- If the image has been found, perform authentication (**HS Device**), and start it (GP/HS).
- If the image has not been found, return from XIP booting to the main booting loop.

Copyright © 2011, Texas Instruments Incorporated

25.3.8.2.1 XIP Initialization and Detection

- GPMC initialization

The following figure and table describes the GPMC timing settings set for XIP boot and other address-data accessible devices (e.g. OneNAND).

Figure 25-9. GPMC XIP Timings

Table 25-8. XIP Timings Parameters

Parameter	Description	Value [clock cycles]
t_{wr}	write cycle period	17
t_{rd}	read cycle period	17
t_{CEon}	CE low time	1
t_{CEoff}	CE high time	16
t_{ADVon}	ADV low time	1
t_{ADVoff}	ADV high time	2
t_{OEon}	OE low time	3
t_{WEon}	WE low time	3
t_{rddata}	data latch time	15
t_{OEoff}	OE high time	16
t_{WEoff}	WE high time	15

The one clock cycle is 20nswhich corresponds to 50-MHz frequency.

- Device detection

There is no specific identification routine executed prior to booting from an XIP device.

25.3.8.2.2 Pins Used

The list of device pins that are configured by the ROM incase of NOR boot mode are as follows. Please note that all the pins might not be driven at boot time. The decision as to which pins need to be driven is done based on the type of NOR flash selected. **On the upper address lines, it is the board designer's responsibility to use necessary pulls so that at POR, the address lines from a12-a27 on the NOR flash are at logic 0**.

Table 25-9. Pins Used for NOR Boot

Signal name	Pin used in XIP_MUX1 mode	Pin used in XIP_MUX2 mode
cs0	gpmc_cs0	gpmc_cs0
advn_ale	gpmc_advn_ale	gpmc_advn_ale
oen_ren	gpmc_oen_ren	gpmc_oen_ren
be0n_cle	gpmc_be0n_cle	gpmc_be0n_cle
wen	gpmc_wen	gpmc_wen
wait	gpmc_wait	gpmc_wait
ad0 - ad15	gpmc_ad0 - gpmc_ad15	gpmc_ad0 - gpmc_ad15
a0	gpmc_a0	lcd_data0
a1	gpmc_a1	**lcd_data1**
a2	**gpmc_a2**	**lcd_data2**
a3	**gpmc_a3**	**lcd_data3**
a4	**gpmc_a4**	**lcd_data4**
a5	gpmc_a5	lcd_data5
a6	**gpmc_a6**	**lcd_data6**
a7	**gpmc_a7**	lcd_data7
a8	**gpmc_a8**	lcd_vsync

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 25-9. Pins Used for NOR Boot (continued)

Signal name	Pin used in XIP_MUX1 mode	Pin used in XIP_MUX2 mode
a9	**gpmc_a9**	lcd_hsync
a10	**gpmc_a10**	lcd_pclk
a11	**gpmc_a11**	lcd_ac_bias_en

25.3.8.2.3 Sysboot Pins

Some of the SYSBOOT pins have special meanings when NOR boot is selected.

Table 25-10. Special SYSBOOT Pins for NOR Boot

SYSBOOT[n]	Description
[8] (CS0BW)	Must be 0 if 8-bit device is usedMust be 1 if 16-bit device is used
[11:10] (CS0MUX[1:0])	00b – Non-muxed device01b – A/D muxed device10b – reserved11b – reserved.

25.3.8.2.4 Image Shadowing for Non-XIP Memories

25.3.8.2.4.1 Shadowing on GP Device

The **GP Device** shadowing uses the following approach.

Figure 25-10. Image Shadowing on GP Device

25.3.8.2.5 NAND

The NAND flash memory is not XIP and requires shadowing before the code can be executed.

25.3.8.2.5.1 Features

- Uses GPMC as the communication interface
- Device from 512Mbit (64MByte)
- x8 and x16 bus width
- Support for large page size (2048 bytes + 64 spare bytes) or very large page size 4096 bytes + 128 / 218 spare bytes)
- Only supports devices where Chip Select can be de-asserted during read, program or erase cycles, without interrupting the operation
- Single Level Cell (SLC) and Multiple Level Cell (MLC) devices
- Device Identification based on ONFI or ROM table
- ECC correction : 8 bits/sector for most devices (16b/sector for devices with large spare area)
- Support for disabling ECC correction, so than the in-built ECC correction mechanisms on some NANDs can be used.
- GPMC timings adjusted for NAND access
- GPMC clock is 50MHz
- Device connected to CS0
- Wait pin signal WAITPIN0 connected to NAND BUSY output
- Four physical blocks are searched for an image. The block size depends on device.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

25.3.8.2.5.2 Initialization and Detection

The initialization routine for NAND devices consists in three parts: GPMC initialization, device detection with parameters determination and finally bad block detection.

ONFI Support

The NAND identification starts with ONFI detection. For more information on ONFI standard, see [19].

GPMC Initialization

The GPMC interface is configured as such it can be used for accessing NAND devices. The address bus is released since a NAND device does not use it. The data bus width is initially set to 8 bits; and changed to 16 bits if needed after device parameters determination. The following scheme is applied since NAND devices require different timings when compared to regular NOR devices:

Figure 25-11. GPMC NAND Timings

Table 25-11. NAND Timings Parameters

Parameter	Description	Value [clock cycles]
t_{wr}	write cycle period	30
t_{rd}	read cycle period	30
t_{CEon}	CE low (not marked on the figure)	0
t_{OEon}	CE low to OE low time	7
t_{WEon}	CE low to WE low time	5
t_{rddata}	CE low to data latch time	21
t_{OEoff}	CE low to OE high time	24
t_{WEoff}	CE low to WE high time	22

Figure 25-11 and Table 25-11 describes the timings configured for NAND device access. The one clock cycle is 20 ns, which correspond to 50-MHz frequency.

Device Detection and Parameters

The ROM Code first performs an initial wait for device auto initialization (with 250ms timeout) with polling of the ready information. Then, it needs to identify the NAND type connected to the GPMC interface. The GPMC is initialized using 8 bits, asynchronous mode. The NAND device is reset (command FFh) and its status is polled until ready for operation (with 200ms timeout). The ONFI Read ID (command 90h / address 20h) is sent to the NAND device. If it replies with the ONFI signature (4 bytes) then a Read parameters page (command ECh) is sent. If the parameters page does not have the ONFI signature, then the ONFI identification fails. If the ONFI identification passes, the information shown in Table 25-12 is then extracted: page size, spare area size, number of pages per block, and the addressing mode. The remaining data bytes from the parameters page stream are simply ignored.

Table 25-12. ONFI Parameters Page Description

Offset	Description	Size (bytes)
6	Features supported	2
80	Number of data bytes per page	4
84	Number of spare bytes per page	2
92	Number of pages per block	4
101	Number of address cycles	1

If the ONFI Read ID command fails (it will be the case with any device not supporting ONFI) then the device is reset again with polling for device to be ready (with 200ms timeout). Then, the standard Read ID (command 90h / address 00h) is sent. If the Device ID (2nd byte of the ID byte stream) is recognized as being a supported device then the device parameters are extracted from an internal ROM Code table. The list of supported devices is shown in Table 25-13.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 25-13. Supported NAND Devices

Capacity	Device ID	Bus Width	Page size
512 Mb	F0	x8	2048
512 Mb	C0	x16	2048
512 Mb	A0	x8	2048
512 Mb	B0	x16	2048
512 Mb	F2	x8	2048
512 Mb	C2	x16	2048
512 Mb	A2	x8	2048
512 Mb	B2	x16	2048
1 Gb	F1	x8	2048
1 Gb	C1	x16	2048
1 Gb	A1	x8	2048
1 Gb	B1	x16	2048
2 Gb	DA	x8	2048
2 Gb	CA	x16	2048
2 Gb	AA	x8	2048
2 Gb	BA	x16	2048
2 Gb	83	x8	2048
2 Gb	93	x16	2048
4 Gb	DC	x8	2048
4 Gb	CC	x16	2048
4 Gb	AC	x8	2048/4096
4 Gb	BC	x16	2048/4096
4 Gb	84	x8	2048
4 Gb	94	x16	2048
8 Gb	D3	x8	2048/4096
8 Gb	C3	x16	2048/4096
8 Gb	A3	x8	2048/4096
8 Gb	B3	x16	2048/4096
8 Gb	85	x8	2048
8 Gb	95	x16	2048
16 Gb	D5	x8	2048/4096
16 Gb	C5	x16	2048/4096
16 Gb	A5	x8	2048/4096
16 Gb	B5	x16	2048/4096
16 Gb	86	x8	2048
16 Gb	96	x16	2048
32 Gb	D7	x8	2048/4096
32 Gb	C7	x16	2048/4096
32 Gb	A7	x8	2048/4096
32 Gb	B7	x16	2048/4096
32 Gb	87	x8	2048
32 Gb	97	x16	2048
64 Gb	DE	x8	2048/4096
64 Gb	CE	x16	2048/4096
64 Gb	AE	x8	2048/4096
64 Gb	BE	x16	2048/4096

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

When the parameters are retrieved from the ROM table: page size and block size is updated based on 4th byte of NAND ID data. Due to inconsistency amongst different manufacturers, only devices which has been recognized to be at least 2Gb (included) have these parameters updated. Therefore, the ROM Code supports 4kB page devices but only if their size, according to the table, is at least 2Gb. Devices which are smaller than 2Gb have the block size parameter fixed to 128kB. Table 25-14 shows the 4th ID Data byte encoding used in ROM Code.

Table 25-14. 4th NAND ID Data Byte

Item	Description	I/O #							
		7	6	5	4	3	2	1	0
Page Size	**1kB**							**0**	**0**
	2kB							0	1
	4kB							1	0
	8kB							1	1
Cell type	2 levels					0	0		
	4 levels					0	1		
	8 levels					1	0		
	16 levels					1	1		
Block Size	64kB			0	0				
	128kB			0	1				
	256kB			1	0				
	512kB			1	1				

Reading NAND Geometry From I2C EEPROM

ROM supports a special boot mode called NANDI2C to support NAND devices whose geometry cannot be detected by the ROM automatically using methods described in the previous section (Figure 12). If this boot mode is selected, the ROM code tries to read NAND geometry from an I2C EEPROM. If the read is successful, ROM code then proceeds to next steps of NAND boot, beginning with reading bad blocks information.

Note: The NAND bus width configuration mentioned in the I2C EEPROM overrides the BUSWIDTH configuration selected by SYSBOOT pins.

The list of device pins that are configured by the ROM incase of NANDI2C boot mode. (This is in addition to the NAND boot pins described in the previous sections.)

Table 25-15. Pins Used for NANDI2C Boot for I2C EEPROM Access

Signal name	Pin Used in Device
I2C SCL	**i2c0_scl**
I2C SDA	**i2c0_sda**

ROM accesses the I2C EEPROM at I2C slave address **0x50** and reads 7 bytes starting from address offset **0x80**. The format of this (NAND geometry information) is as follows:

Table 25-16. NAND Geometry Information on I2C EEPROM

Byte address	Information	
	Upper nibble	Lower nibble
0x80	Magic Number – 0x10	
0x81	Magic Number – 0xb3	
0x82	Magic Number – 0x57	
0x83	Magic Number – 0xa6	
0x84	NAND column address (word/byte offset within a page) size in bytes, Example: 2	NAND row address (page offset) size in bytes. Example: 3

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 25-16. NAND Geometry Information on I2C EEPROM (continued)

Byte address	Information	
0x85	Page size (2^N) exponent "N". Example (for page size of 2048): 11	Pages per block (2^N) exponent "N"Example (for number of blocks 64): 6
0x86	NAND bus width0 è 8-bit, 1 è 16-bit	ECC Type0 è No ECC, 1 è BCH8, 2 è BCH16

- ECC correction

The default ECC correction applied is BCH 8b/sector using the GPMC and ELM hardware.

For device ID codes D3h, C3h, D5h, C5h, D7h, C7h, DEh, CEh when manufacturer code (first ID byte) is 98h the Cell type information is checked in the 4th byte of ID data. If it is equal to 10b then the ECC correction applied is BCH 16b/sector.

In addition ECC computation done by the ROM can be turned off completely by using SYSBOOT[9]. This is particularly useful when interfacing with NAND devices that have built in ECC engines.

Table 25-17. ECC Configuration for NAND Boot

SYSBOOT[9]	ECC Computation
0	ECC done by ROM
1	ECC handled by NAND

Figure 25-12. NAND Device Detection

The detection procedure is described in Figure 25-12. Once the device has been successfully detected, the ROM Code changes GPMC to 16-bit bus width if necessary.

Bad Block Verification

Invalid blocks are blocks which contain invalid bits whose reliability cannot be guaranteed by the manufacturer. Those bits are identified in the factory or during the programming and reported in the initial invalid block information located in the spare area on the 1st and 2nd page of each block. Since the ROM Code is looking for an image in the first four blocks, it must detect block validity status of these blocks. Blocks which are detected as invalid are not accessed later on. Blocks validity status is coded in the spare areas of the first two pages of a block (first byte equal to FFh in 1st and 2nd pages for an 8 bits device / first word equal to FFFFh in 1st and 2nd page for a 16bits device).

Figure 25-13 depicts the invalid block detection routine. The routine consists in reading spare areas and checking validity data pattern.

Figure 25-13. NAND Invalid Blocks Detection

25.3.8.2.5.2.1 NAND Read Sector Procedure

The ROM Code reads data from NAND devices in 512 bytes sectors. The read function fails in two cases:

- The accessed sector is within a block marked as invalid
- The accessed sector contains an error which cannot be corrected with ECC

Figure 25-14 shows the read sector routine for NAND devices. The ROM Code uses normal read (command 00h 30h) for reading NAND page data.

Figure 25-14. NAND Read Sector Procedure

Page data can contain errors due to memory alteration. The ROM Code uses an ECC correction algorithm to detect and possibly correct those errors. The ECC algorithm used is BCH with capability for correcting 8b or 16b errors per sector. The BCH data is automatically calculated by the GPMC on reading each 512 bytes sector. The computed ECC is compared against ECC stored in the spare area for the corresponding page. Depending on the page size, the amount of ECC data bytes stored in the corresponding spare area

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

is different. Figure 25-15 and Figure 25-16 show the mapping of ECC data inside the spare area for respectively 2KB-page and 4KB- page devices. If both ECC data are equal then the Read Sector function returns the read 512 bytes sector without error. Otherwise the ROM Code tries to correct error(s) in the corresponding sector (this procedure is assisted by the ELM hardware) and returns the data if successful. If errors are uncorrectable, the function returns with FAIL.

Figure 25-15. ECC Data Mapping for 2 KB Page and 8b BCH Encoding

Figure 25-16. ECC Data Mapping for 4 KB Page and 16b BCH Encoding

25.3.8.2.5.2.2 Pins Used

The list of device pins that are configured by the ROM incase of NAND boot mode are as follows. Please note that all the pins might not be driven at boot time.

Table 25-18. Pins Used for NAND Boot

Signal name	Pin Used in Device
cs0	gpmc_cs0
advn_ale	gpmc_advn_ale
oen_ren	gpmc_oen_ren
be0n_cle	gpmc_be0n_cle
wen	gpmc_wen
wait	gpmc_wait
clk	gpmc_clk
ad0 - ad15	gpmc_ad0 - gpmc_ad15

25.3.8.2.5.2.3 Sysboot Pins

Some of the SYSBOOT pins have special meanings when NAND boot is selected.

Table 25-19. Special SYSBOOT Pins for NAND Boot

SYSBOOT[n]	Description
[8] (CS0BW)	Must be 0 if 8-bit NAND is usedMust be 1 if 16-bit NAND is used
[9] (CS0WAIT)	Must be 0
[11:10] (CS0MUX[1:0])	Must be 0

25.3.8.2.6 MMC / SD Cards

25.3.8.2.6.1 Overview

The ROM Code supports booting from MMC / SD cards in the following conditions:

- MMC/SD Cards compliant to [6], [9] of low and high capacities.
- MMC/SD cards connected to MMC interface #0 and #1.
- 3V VCC power supply, support for 3- or 1.8-V I/O voltages.
- Initial 1-bit MMC Mode, optional 4-bit mode, if device supports it.
- Clock Frequency: identification mode: 400 KHz; data transfer mode up to 10 MHz.
- Only one card connected to the bus.
- Raw mode, image data read directly from sectors in the user area.
- File system mode (FAT12/16/32 supported with or without Master Boot Record), image data is read from a booting file.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

25.3.8.2.6.2 System Interconnection

An MMC/SD card can be connected to the **MMC0 interface** typically through a card cage.

An eMMC/eSD/managed NAND memory device can be connected to **MMC1 interface**.

Note:

- The ROM Code does not handle the card detection feature on card cage.
- If MMC1 is used the GPMC interface is not usable, due to pin muxing options.
- MMC1 supports sector mode without querying the card.

25.3.8.2.6.3 Booting Procedure

The high level flowchart of the eMMC / eSD and MMC/SD booting procedure is depicted in Figure 25-17. The booting file is searched only in case of booting from a card. eMMC/eSD embedded memories only support raw mode.

Figure 25-17. MMC/SD Booting

25.3.8.2.6.4 Initialization and Detection

The ROM Code initializes the memory device or card connected on MMC interface using the standard High-Voltage range (3.0V). If neither memory device nor card is detected then the ROM Code carries on to the next booting device. The standard identification process and **Relative Card Address (RCA)** assignment are used. However, the ROM Code assumes that only one memory or card is present on the bus. This first sequence is done using the CMD signal which is common to SD and MMC devices.

MMC and SD standards detail this phase as **initialization phase**. Both standards differ in the first commands involved: **CMD1** and **ACMD41**. The ROM Code uses this difference in command set to discriminate between MMC and SD devices: CMD1 is supported only by the MMC standard whereas ACMD41 is only supported by SD standard. The ROM Code first sends a CMD1 to the device and gets a response only if an MMC device is connected. If no response is received then ACMD41 (ACMD41 is made out of CMD55 and ACMD41) is sent and a response is expected from an SD device. If no response is received then it is assumed that no device is connected and the ROM Code exits the MMC/SD Booting procedure with FAIL. This is detection procedure shown in Figure 25-18.

Another point to note is the difference in the arguments to CMD1 between MC0 and MMC1. At first the ROM queries the card with CMD1, ARG = 0, to get the OCR from the card. On MMC0, the response from the card is simply reflected back to the card as the argument for all subsequent CMD1, till the card is in the READY state. On MMC1, Bit30 of the response received from the card is set to 1 by the ROM, and this modified value is used as the argument for subsequent CMD1. This is done to indicate to the card that the ROM supports sector addressing. This mode might not be compatible with older (older than v4.4) versions of cards.

Figure 25-18. MMC/SD Detection Procedure

As previously mentioned the contents of an MMC/SD card may be formatted as raw binary or within a FAT filesystem.. eMMC / eSD devices only support raw mode. The ROM Code reads out raw sectors from image or the booting file within the file system and boots from it.

25.3.8.2.6.5 MMC/SD Read Sector Procedure in Raw mode

In raw mode the booting image can be located at one of the four consecutive locations in the main area: offset 0x0 / 0x20000 (128 KB) / 0x40000 (256 KB) / 0x60000 (384 KB). For this reason, a booting image shall not exceed 128KB in size. However it is possible to flash a device with an image greater than 128KB starting at one of the aforementioned locations. Therefore the ROM Code does not check the image size. The only drawback is that the image will cross the subsequent image boundary.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

The raw mode is detected by reading sectors #0, #256, #512, #768. The content of these sectors is then verified for presence of a TOC structure as described in Section 25.3.10. In the case of a **GP Device**, a Configuration Header (CH) **must** be located in the first sector followed by a **GP header**. The CH might be void (only containing a CHSETTINGS item for which the Valid field is zero). In the case of an **HS Device** the TOC must contain an "MLO" item.

25.3.8.2.6.6 MMC/SD Read Sector Procedure in FAT Mode

MMC/SD Cards may hold a FAT file system which ROM Code is able to read and process. The image used by the booting procedure is taken from a specific booting file named "MLO". This file has to be located in the root directory on an active primary partition of type FAT12/16 or FAT32. See [8] and [10] for a more detailed description of MMC/SD file system support.

An MMC/SD card can be configured either as floppy-like or hard-drive-like.

- When acting as floppy-like, the content of the card is a single file system without any Master Boot Record (MBR) holding a partition table
- When acting as hard-drive-like, an MBR is present in the first sector of the card. This MBR holds a table of partitions, one of which must be FAT12/16/32, primary and active.

According to [8], the card should always hold an MBR except for MMC cards using floppy-like file system (please refer to the CSD internal Register fields FILE_FORMAT_GRP and FILE_FORMAT in [6]). However, depending on the used operating system the MMC/SD card will be formatted either with partition(s) (using an MBR) or without. The ROM Code supports both types; this is described in the following section.

The ROM Code retrieves a map of the booting file from the FAT table. The booting file map is a collection of all FAT table entries related to the booting file (a FAT entry points to a cluster holding part of the file). The booting procedure uses this map to access any 512 byte sector within the booting file without involving ROM Code FAT module.

The sector read procedure utilizes standard MMC/SD raw data read function. The sector address is generated based on the booting memory file map collected during the initialization. Hence the ROM Code can address sectors freely within the booting file space.

25.3.8.2.6.7 FAT File system

This paragraph describes functions which are used by the ROM Code, it is not intended to fully describe the Master Boot Record and the FAT file system:

- How to recognize if a sector is the 1st sector of an MBR
- How to recognize if a sector is the 1st sector of a FAT12/16/32
- How to find the 1st cluster of the booting file
- How to buffer the booting file FAT entries.

Some memory devices which support file systems can be formatted with or without MBR, therefore the first task of the ROM Code is to detect whether or not the device is holding an MBR in the first sector.

If this is the case, an active FAT12/16/32 partition is searched in all 4 MBR partition entries, based on the Type field. If the MBR entries are not valid or if no useable partition is found then the ROM Code returns to the Booting procedure with FAIL. The Extended partitions are not checked, the booting file must reside in a primary partition.

If a partition is found then its first sector is read and used further on. If no MBR is present (in case of a floppy-like system), the first sector of the device is read and used further on.

The read sector is checked to be a valid FAT12/16 or FAT32 partition. If this fails, in case another partition type is used (i.e. Linux FS or any other) or if the partition is not valid, the ROM Code returns with FAIL.

Otherwise, the Root Directory entries are searched for a file named depending on the booting device. The Long File Names (LFN) format is not used and **only File Names in 8.3 Format are searched for**. If no valid file is found, the ROM Code returns with FAIL.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Once the file has been found, the ROM Code reads the File Allocation Table (FAT) and buffers the singly-linked chain of clusters in a FAT Buffer which will be used by the Booting Procedure to access the file directly sector by sector. For FAT12/16 and for FAT32 (valid if a specific flag has been set in the FAT32 Boot Sector), there exist multiples copies of the FAT (ROM Code supports only 2 copies). When buffering FAT entries, the 2 FATs are compared. If they are not the same, only entries from the last FAT are used. The FAT Buffer holds sector numbers and not cluster numbers. The ROM Code converts each cluster entry to one or several sector entries if applicable.

The whole process is described in Figure 25-19. Every part related to MBR or FAT12/16/32 is described in the next paragraph.

Figure 25-19. SD/MMC Booting, Get Booting File

25.3.8.2.6.7.1 Master Boot Record (MBR)

The Master Boot Record is the 1st sector of a memory device. It is made out of some executable code and 4 partition entries. The aim of such a structure is to be able to divide the hard disk in partitions mostly used to boot different systems (i.e. Microsoft Windows™, Linux, …). Its structure is described in Table 25-20 and Table 25-21. The valid partition types searched by the ROM Code are described in Table 25-22.

Table 25-20. Master Boot Record Structure

Offset	Length[bytes]	Entry Description	Value
0000h	446	Optional Code	
01BEh	16	Partition Table Entry	(see Table 25-21)
01CEh	16	Partition Table Entry	(see Table 25-21)
01DEh	16	Partition Table Entry	(see Table 25-21)
01EEh	16	Partition Table Entry	(see Table 25-21)
01FEh	2	Signature	AA55h

Table 25-21. Partition Entry

Offset	Length[bytes]	Entry Description	Value
0000h	1	Partition State	00h: Inactive80h: Active
0001h	1	Partition Start Head	H_s
0002h	2	Partition Start Cylinder and Sector	$C_s[7:0]$-$C_s[9:8]$-$S_s[5:0]$
0004h	1	Partition Type	See Table 25-22 for partial partition types
0005h	16	Partition End Head	H_e
0006h	2	Partition End Cylinder and Sector	$C_e[7:0]$-$C_e[9:8]$-$S_e[5:0]$
0008h	4	First sector position relative to the beginning of media	$LBA_s=C_s.H.S+ H_s.S+ S_s-1$
000Ch	4	Number of sectors in partition	$LBA_e=C_e.H.S+ H_e.S+ S_e-1Nbs= LBA_e-LBA_s+1$

Table 25-22. Partition Types

Partition Type	Description
01h	FAT12
04h, 06h, 0Eh	FAT16
0Bh, 0Ch, 0Fh	FAT32

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

The way the ROM Code detects whether a sector is the 1st sector of an MBR or not is described in Figure 25-20.

The ROM Code first checks if the signature is present. Each partition entry is checked:

- If its type is set to 00h then all fields in the entry must be 00h
- The partition is checked to be within physical boundaries, i.e. the partition is located inside and it's size fits the total physical sectors.

Figure 25-20. MBR Detection Procedure

Once identified, the ROM Code gets the partition using the procedure described in Figure 25-21. The partition type is checked to be FAT12/16 or FAT32. Its state must be 00h or 80h (if there are more than one active partition the test fails). The ROM Code returns with FAIL if no active primary FAT12/16/32 could be found.

Figure 25-21. MBR, Get Partition

25.3.8.2.6.7.2 FAT12/16/32 Boot Sector

The FAT file system is made out of several parts:

- Boot Sector which holds the BIOS Parameter Block (BPB)
- File Allocation Table (FAT) which describes the use of each cluster of the partition
- Data Area which holds the Files, Directories and Root Directory (for FAT12/16, the Root Directory has a specific fixed location).

The Boot Sector is described in Table 25-23. For more details, see [12].

Note: In the following description, all the fields whose names start with BPB_ are part of the BPB. All the fields whose names start with BS_ are part of the Boot Sector and not really part of the BPB (not mandatory), they are not used at all by the ROM Code.

Table 25-23. FAT Boot Sector

Offset	Length[bytes]	Name	Description
0000h	3	BS_jmpBoot	Jump Instruction to Boot Code (not used)
0003h	8	BS_OEMName	Name of the System which created the partition
000Bh	2	BPB_BytsPerSec	Counts of Bytes per sector (usually 512)
000Dh	1	BPB_SecPerClus	Number of sectors per allocation unit
000Eh	2	BPB_RsvdSecCnt	Number of reserved sectors for the Boot SectorFor FAT12/16 is 1, for FAT32, usually 32
0010h	1	BPB_NumFATs	Number of copies of FAT, usually 2
0011h	2	BPB_RootEntCnt	For FAT12/16, number of 32 bytes entries in the Root Directory (multiple of BPB_BytsPerSec/32)For FAT32 this value is 0
0013h	2	BPB_TotSec16	Total Count of sectors on the volume. If the size is bigger than 10000h or for FAT32, this field is 0 and BPB_TotSec32 holds the value
0015h	1	BPB_Media	Media Type, usually F8h: fixed, non-removable
0016h	2	BPB_FATSz16	For FAT12/16, size in sectors of one FATFor FAT32, holds 0
0018h	2	BPB_SecPerTrk	Number of sectors per track, 63 for SD/MMC
001Ah	2	BPB_NumHeads	Number of heads, 255 for SD/MMC

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 25-23. FAT Boot Sector (continued)

Offset		Length[bytes]	Name	Description	
001Ch		4	BPB_HiddSec	Number of sectors preceeding the partition	
0020h		4	BPB_TotSec32	Total Count of sectors on the volume. If the size is smaller than 10000h (for FAT12/16), this field is 0 and BPB_TotSec16 is valid	
FAT12/16	0024h	1	BS_DrvNum	Drive Number	
	0025h	1	BS_Reserved1	00h	
	0026h	1	BS_BootSig	Extended Boot Signature 29h. Indicates that the following 3 fields are present	
	0027h	4	BS_VolID	Volume Serial Number	
	002Bh	11	BS_VolLab	Volume Label	
	0036h	8	BS_FilSysType	File system Type: “FAT12”, “FAT16”, “FAT32”.Note: This field is not mandatory (i.e BS_) therefore it cannot be used to indentify the partition type.	
FAT32	0024h	4	BPB_FATSz32	Size in sectors of one FAT. Field BPB_FATSz16 must be 0	
	0028h	2	BPB_ExtFlags	FAT Flags:	
				[7]:	0=FAT is mirrored; 1=Only one FAT is used
				[3:0]:	Number of used FAT if no mirroring used
	002Ah	2	BPB_FSVer	File system Version Number	
	002Ch	4	BPB_RootClus	First Cluster number of the Root Directory	
	0030h	2	BPB_FSInfo	Sector number of FSINFO Structure in the reserved-area, usually 1	
	0032h	2	BPB_BkBootSec	If non-zero, indicates the sector number in the reserved-area of a copy of the Boot Sector	
	0034h	12	BPB_Reserved	Reserved, set to 00h	
	0040h	1	BS_DrvNum	Drive Number	
	0041h	1	BS_Reserved1	00h	
	0042h	1	BS_BootSig	Extended Boot Signature 29h. Indicates that the following 3 fields are present	
	0043h	4	BS_VolID	Volume Serial Number	
	0047h	11	BS_VolLab	Volume Label	
	0052h	8	BS_FilSysType	File system Type: “FAT12”, “FAT16”, “FAT32”.Note: This field is not mandatory (i.e BS_) therefore it cannot be used to indentify the partition type.	
01FEh		2	BPB_Signature	AA55h	

To check whether or not a sector holds a valid FAT12/16/32 partition, only fields starting with BPB can be checked as they are mandatory. The fields starting from offset 0024h to 01FDh cannot be used for the check as they will differ if using FAT12/16 or FAT32. The procedure is described in Figure 30. First the ROM Code checks if the *BPB_Signature* is equal to AA55h. Then it checks some fields which must have some specific values (*BPB_BytsPerSec, BPB_SecPerClus, BPB_RsvdSecCnt, BPB_NumFATs, BPB_RootEntCnt*) If the geometry of the device is known (valid CHS for device size < 4Gbytes) then it is compared against *BPB_SecPerTrk* and *BPB_NumHeads* fields. If an MBR was found before, the partition size is also checked:

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

The field *BPB_ToSec16* is used if the total number of sectors is below 65518 (in this case *BPB_TotSec32*=0), otherwise *BPB_TotSec32* is used (*BPB_TotSec16*=0). The partition sector offset is also checked: *BPB_HiddSec = MBR_Partition_Offset* (if this value is not 0 as some operating systems do not update this field correctly). The last step is to decide which type of FAT file system it is. The ROM Code computes the number of clusters in the Data Area part of the partition:

Where *Nb clusters* is given by the size of the Data Area:

Figure 25-22. FAT Detection Procedure

25.3.8.2.6.7.3 FAT12/16/32 Root Directory

The next task for the ROM Code is to find the booting file named “MLO” inside the Root Directory of the FAT12/16/32 file system. The file is not searched in any other location.

For a FAT12/16 file system, the Root Directory has a fixed location which is cluster 0. For a FAT32 file system, its cluster location is given by *BPB_RootClus*. The general formulae to find the sector number (relative to device sector 0, not partition sector 0) of a cluster is given by:

$$Cluster_{sector} = BPB_RsvdSecCnt + BPB_NumFATs \times BPB_FATSz + Cluster \times BPB_SecPerClus$$

Note: BPB_FatSz is BPB_FatSz16 for FAT12/16 or BPB_FatSz32 for FAT32

Note: the *BPB_HiddSec* field can contain 0 even though the FAT file system is located somewhere other than on sector 0 (floppy-like). The ROM Code actually uses the partition offset taken from the MBR instead of this field which can be wrong. If no MBR was found (floppy-like) the value 0 is used.

Each entry in the Root Directory is 32 bytes long and hold information about the file, i.e. filename, date of creation, rights, cluster location etc. This is described in Table 25-24.

The ROM Code checks each entry in the Root Directory until either the booting file is found or the entry is empty (first byte is 00h) or when the end of the Root Directory has been reached. Entries with ATTR_LONG_NAME attribute (LFN) and with first byte at E5h (erased file) are ignored. When found, the first cluster offset of the file is read from the DIR_FstClusHi/DIR_FstClusLo fields.

There is a slight difference between FAT12/16 and FAT32 when handling the Root Directory. On FAT12/16, this directory has a fixed location (see above) and length fixed by *BPB_RootEntCnt* which is the total number of 32 bytes entries. Handling this directory is therefore straight forward. On FAT32, the Root Directory is like a standard file, the File Allocation Table (FAT) has to be used in order to retrieve each sector of the Directory. The way the FAT is handled is described in the next paragraph.

Table 25-24. FAT Directory Entry

Offset	Length[bytes]	Name	Description	
0000h	11	DIR_Name	Short Name (8+3)	
000Bh	1	DIR_Attr	File Attributes:	
			ATTR_READ_ONLY	01h
			ATTR_HIDDEN	02h
			ATTR_SYSTEM	04h
			ATTR_VOLUME_ID	08h
			ATTR_DIRECTORY	10h
			ATTR_ARCHIVE	20h
			ATTR_LONG_NAME	ATTR_READ_ONLY | ATTR_HIDDEN | ATTR_SYSTEM | ATTR_VOLUME_ID
000Ch	1	DIR_NTRes	Reserved, set to 00h	
000Dh	1	DIR_CrtTimeTenth	Millisecond stamp at file creation	
000Eh	2	DIR_CrtTime	Time file was created	
0010h	2	DIR_CrtDate	Date file was created	
0012h	2	DIR_LstAccDate	Last Access date	

Copyright © 2011, Texas Instruments Incorporated

Table 25-24. FAT Directory Entry (continued)

Offset	Length[bytes]	Name	Description
0014h	2	DIR_FstClusHi	High word of this entry's first cluster number
0016h	2	DIR_WrtTime	Time of last write
0018h	2	DIR_WrtDate	Date of last write
001Ah	2	DIR_FstClusLo	Low word of this entry's first cluster number
001Ch	4	DIR_FileSize	File size in bytes

25.3.8.2.6.7.4 FAT12/16/32 File Allocation Table

The ROM Code needs to read the FAT in order to retrieve sectors either for the booting file or for the Root Directory (in case the file system is FAT32).

There can be multiple copies of the FAT inside the file system (ROM Code supports only 2) located after the Boot Sector:

$FATn_{sector} = BPB_HiddSec+BPB_RsvdSecCnt+BPB_FatSz \times n$

Its size is given by *BPB_FATSz16* or *BPB_FATSz32*. The ROM Code checks each copy of the FAT if identical. In case the values are different, the ROM Code uses the value from the last FAT copy.

With FAT32 file system, the copy system can be disabled according to a flag located in *BPB_ExtFlags[7]*. If this flag is set then FAT *BPB_ExtFlags[3:0]* is used. In this case no verification is made by the ROM Code with other copies of FAT.

The FAT is a simple array of values each refering to a cluster located in the Data Area. One entry of the array is 12, 16 or 32 bits depending on the file system in use.

The value inside an entry defines whether the cluster is being used or not and if another cluster has to be taken into account. This creates a singly-linked chain of clusters defining the file. The meaning of an entry is described in Table 25-25.

Note: For compatibility reasons, clusters 0 and 1 are not used for files and those entries must contain FF8h and FFFh (for FAT12); FFF8h and FFFFh (for FAT16); ?FFFFFF8h and ?FFFFFFFh (for FAT32).

Table 25-25. FAT Entry Description

FAT12	FAT16	FAT32	Description
000h	0000h	?0000000h	Free Cluster
001h	0001h	?0000001h	Reserved Cluster
002h-FEFh	0002h-FFEFh	00000002h-?FFFFFEFh	Used Cluster; value points to next cluster
FF0h-FF6h	FFF0h-FFF6h	?FFFFFF0h-?FFFFFF6h	Reserved values
FF7h	FFF7h	?FFFFFF7h	Bad Cluster
FF8h-FFFh	FFF8h-FFFFh	?FFFFFF8h-?FFFFFFFh	Last Cluster in File

Note: FAT32 uses only bits [27:0], the upper 4 bits are usually 0 and should be left untouched.

When accessing the Root Directory for FAT32, the ROM Code just starts from the Root Directory Cluster entry and follows the linked chain to retrieve the clusters.

When the booting file has been found, the ROM Code buffers each FAT entry corresponding to the file in a sector way. This means each cluster is translated to one or several sectors depending on how many sectors are in a cluster (*BPB_SecPerClus*). This buffer is used later on by the booting procedure to access the file.

25.3.8.2.6.8 Pins Used

The list of device pins that are configured by the ROM incase of MMC boot mode are as follows. Please note that all the pins might not be driven at boot time.

Copyright © 2011, Texas Instruments Incorporated

Table 25-26. Pins Used for MMC0 Boot

Signal name	Pin Used in Device
clk	**mmc0_clk**
cmd	**mmc0_cmd**
dat0	**mmc0_dat0**
dat1	**mmc0_dat1**
dat2	**mmc0_dat2**
dat3	**mmc0_dat3**

Table 25-27. Pins Used for MMC1 Boot

Signal name	Pin Used in Device
clk	gpmc_csn1
cmd	gpmc_csn2
dat0	gpmc_ad0
dat1	gpmc_ad1
dat2	gpmc_ad2
dat3	gpmc_ad3

25.3.8.2.7 SPI

SPI EEPROMs or SPI flashes have an EEPROM or NOR flash backend and they connect to the device using the serial SPI protocol.

These typically devices operate in three stages – the command stage, the address stage and the data transfer stage. The command is usually an 8 bit value followed by the address (depending on the size of the device) followed by the data to be read or written.

Because of the need for fewer pins, these devices are comparatively inexpensive, easy for board layout, and are the devices of choice when cost, complexity and form factor are critical considerations.

25.3.8.2.7.1 Features

- Supports 12-MHz clock (50% duty cycle)
- Supports only SPI Mode 3 (clock polarity = 1, clock phase = 1)
- Supports only 24 bit addressable EEPROMs
- Supports only 4-pin SPI mode (CS, CLK, Serial Input, Serial Output)
- The boot devices must be connected to chip select 0 and must support the read command (03h)
- The boot image is copied into internal memory and then executed.

25.3.8.2.7.2 Initialization and Detection

The ROM Code initializes the SPI controller, pin muxing and clocks for communicating with the SPI device. The controller is initialized in Mode 3 and the clock is setup to operate at 12 MHz. There is no specific device identification routine that is executed by the ROM code to identify whether a boot device is preset or not. If no SPI device is present, the sector read will return only 0xFFFFFFFF and the SPI boot will be treated as failed.

25.3.8.2.7.3 SPI Read Sector Procedure

The ROM Code reads SPI data from the boot device in 512 byte sectors. For each call to the SPI Read Sector routine, the SPI Read Command (0x03) is sent along with the 24 bit start address of the data to be read.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Form the next iteration onwards, a dummy value is transmitted on the master out line and the data is received on the master in line. This needs to be done because SPI protocol always operated in full duplex mode. The dummy data transmitted by the ROM is the Read Command appended to the start address. The data from the boot device is received MSB first.

As the A8 is a little-endian processor, and SPI operates in a big-endian format, this means that while writing to the flash, care needs to be taken to write the image in a big-endian format. This way we can avoid doing the endian conversion at boot time, thus improving boot performance.

25.3.8.2.7.4 Pins Used

The list of device pins that are configured by the ROM incase of SPI boot mode are as follows. Please note that all the pins might not be driven at boot time.

Table 25-28. Pins Used for SPI Boot

Signal name	Pin Used in Device
cs	spi0_cs0
miso	spi0_d0
mosi	spi0_d1
clk	spi0_sclk

25.3.8.3 Blocks and Sectors Search Summary

Table 25-29 summarizes numbers of blocks and sectors which are searched during the memory booting from devices requiring image shadowing. NAND is organized with blocks, which are erasable units.

Table 25-29. Blocks and Sectors Searched on Non-XIP Memories

Memory	Maximum Number of Blocks Checked	Number of Sectors Searched
NAND	first 4	Number of sectors in one block[(1)]
SPI, eMMC/eSD and MMC/SD cards (raw mode)	first 4	1

(1) Depends on NAND geometry

Regarding MMC/SD card booting in FAT mode, the file system area is searched for one file.

25.3.9 Peripheral Booting

25.3.9.1 Overview

The ROM Code can boot from four different peripheral interfaces:

- EMAC: 1000/100/10 Mbps Ethernet, using standard TCP/IP network boot protocols BOOTP and TFTP
- PCIe: Two lanes @250 MB/s
- USB: Full speed, client mode
- UART: 115.2Kbps, 8 bits, even parity, 1 stop bit, no flow control

The purpose of booting from a **peripheral interface** is to download a **boot image** from an external host (typically a PC). This booting method is mostly used for programming flash memories connected to the device (e.g. in the case of initial flashing, firmware update or servicing).

25.3.9.2 Boot Image Location and Size

The boot image is downloaded directly into internal RAM at the location 0x402F0000 on GP devices and 0x40300000h on HS devices. The maximum size of downloaded image is 110 KB.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

25.3.9.3 Peripheral Boot Procedure Overview

Figure 25-23. Peripheral Booting Procedure

25.3.9.4 EMAC Boot Procedure

25.3.9.4.1 Device Initialization

- EMAC boot uses the CPGMAC port 1 of the device.
- Supports connection to external Ethernet PHY using the MII, RMI, RGMII and MDIO pins, based on SYSBOOT pin settings.
- Device uses EFUSE register MAC_ID0 for Ethernet MAC address of the device.
- Device detects if the PHY is alive on the MDIO interface and
  - Reads the STATUS register to check if Ethernet link is active
  - Reads the CONTROL register to detect the auto-negotiated mode of operation
    - Is the mode full-duplex or half duplex
    - Speed of operation, 1000/100/10 Mbps.
  - Waits for 3.5s for auto negotiation to complete before timing out.

25.3.9.4.2 BOOTP (RFC 951)

The device then proceeds to obtain the IP and Boot information using BOOTP protocol. The device prepares and broadcasts the BOOTP message that has the following information:

- Device MAC address in “chaddr” field – to uniquely identify the device to the server.
- “vender-class-identifier” option number 60 (RFC 1497, RFC 1533). Servers could use this information to identify the device type. The value present is "**DM816x ROM v1.0**"
- “”Client-identifier” option number 61 (RFC 1497, RFC 1533). This has the **ASIC-ID** structure which contains additional info for the device.

The device then expects a BOOTP response that provides the following information for the booting to proceed

- Device IP address from “yiaddr” field
- Subnetmask from extended option 1 (RFC 1497, RFC 1533)
- Gateway IP from extended option number 3 (RFC 1497, RFC 1533) or from “giaddr” field of BOOTP response.
- Boot image filename from “file” field
- TFTP server IP address from the “siaddr” field

Timeouts and retries

- Exponentially increasing timeouts starting from 4s, doubling for each retry.
- 5 retries

25.3.9.4.3 TFTP (RFC 1350)

After a successful BOOTP completion, the device initiates the TFTP download of the boot image into SRAM. The device has the capability to reach TFTP server within the local subnet or outside, though the gateway.

Timeouts and retries:

- Timeout of 1s to receive a response for the READ request
- 5 retries for the READ request
- Retries are managed by server once data transfer starts (server re-sends a data packet if the ACK was not received within a timeout value)
- Device has a 60s timeout to complete the data transfer, to handle the scenario if the server dies in the

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

middle of a data transfer

25.3.9.4.4 Pins Used

The list of device pins that are configured by the ROM incase of EMAC boot mode are as follows. Please note that all the pins might not be driven at boot time.

Table 25-30. Pins Used for EMAC Boot in MII Mode

Signal name	Pin Used in Device
col	**mii1_col**
crs	**mii1_crs**
gtx_clk	**mii1_gtxclk**
rx_clk	**mii1_rxclk**
rxd0-rxd3	**mii1_rxd0-rxd3**
rxdv	**mii1_rxdv**
rxer	**mii1_rxer**
tx_clk	**mii1_txclk**
txd_0-txd3	**mii1_txd0-txd3**
txen	**mii1_txen**
mclk	mdio_clk
mdio	mdio_data

Table 25-31. Pins Used for EMAC Boot in RGMII Mode

Signal name	Pin Used in Device
rxclk	mii1_rxclk
rxctl	**mii1_rxdv**
txclk	**mii1_txclk**
txctl	**mii1_txen**
rxd0	**mii1_rxd0 – rxd3**
txd0	**mii1_txd0 –txd3**
mclk	mdio_clk
mdio	mdio_data

Table 25-32. Pins Used for EMAC Boot in RMII Mode

Signal name	Pin Used in Device
refclk	**rmii1_refclk (driven by SOC)**
crs	mii1_crs
rxer	**mii1_rxerr**
txen	mii1_txen
rxd0	**mii1_rxd0 – rxd1**
txd0	**mii1_txd0 – txd1**
mclk	mdio_clk
mdio	mdio_data

25.3.9.4.5 Sysboot pins

Some of the SYSBOOT pins have special meanings when EMAC boot is selected.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 25-33. Ethernet PHY Mode Selection

SYSBOOT[7:6]	PHY Mode
00b	MII
01b	RMII
10b	RGMII with internal delay
11b	RGMII without internal delay

25.3.9.5 UART Boot Procedure

25.3.9.5.1 Device Initialization

- UART boot uses UART0.
- UART0 is configured to run at 115200 baud, 8-bits, even parity, 1 stop bit and no flow control.

25.3.9.5.2 Boot Image Download

- UART boot uses x-modem client protocol to receive the boot image.
- Utilities like hyperterm, teraterm, minicom can be used on the PC side to download the boot image to the board
- With x-modem packet size of 1K throughout is roughly about 4KBytes/Sec.
- The ROM code will ping the host 10 times in 3s to start x-modem transfer. If host does not respond, UART boot will timeout.
- Once the transfer has started, if the host does not send any packet for 3s, UART boot will time out
- If the delay between two consecutive bytes of the same packet is more than 2ms, the host is requested to re-transmit the entire packet again
- Error checking using the CRC-16 support in x-modem. If an error is detected, the host is requested to re-transmit the packet again.

25.3.9.5.3 Pins Used

The list of device pins that are configured by the ROM incase of UART boot mode are as follows. Please note that all the pins might not be driven at boot time.

Table 25-34. Pins Used for UART Boot

Signal name	Pin Used in Device
rx	uart0_rxd
tx	uart0_txd

25.3.9.6 USB Boot Procedure

25.3.9.6.1 Device Initialization

The ROM Code supports booting from the USB interface under the following conditions:

- Using the High Speed USB OTG (USBOTGHS) IP through USB0 interface.
- USB operates in Full Speed, Client mode.
- USB will operate only in device powered mode.
- **Integrated transceiver** (through UTMI).
- The enumeration default timeout is **3s** (Timeout Usb).

Even though using an OTG capable hardware, the ROM Code does not handle any OTG specific feature.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

25.3.9.6.1.1 Overview

In case of boot from USB is chosen by the SYSBOOT pin configuration:

- The USBOTGHS hardware and PRCM clocks are configured for UTMI mode
- The ROM Code continues with the USB procedure only if the USB cable is detected present (i.e. VBUS is detected at transceiver level and communicated as such through the UTPI traffic). If not, the initialization procedure is aborted.
- The ROM code implements the RNDIS class driver
- From users perspective, USB boot is indistinguishable from Ethernet boot.
- The USB initialization procedure is shown in Figure 25-24.

Figure 25-24. USB Initialization Procedure

25.3.9.6.1.2 Enumeration Descriptors

The Device Descriptor parameters which are used during enumeration are listed in the following table. The default Vendor ID and Product ID can be automatically overridden by the customer by programming the Efuses that are used to store these values.

Table 25-35. Customized Descriptor Parameters

Parameter	Size [bytes]	TI Default Values
Device ID code	2	0000h
Device Class	1	02h
Device Sub-Class	1	FFh
Device Protocol	1	FFh
Manufacturer	String	“Texas Instruments”
Product	String	“USBRNDIS”
Serial number	String	“N/A”

.

25.3.9.6.2 Image Download Procedure

- The device supports USB client boot only.
- The ROM implements as RNDIS class driver, so the device enumerates as an ethernet port.
- Standard RNDIS drivers present on Linux and Windows are picked up during the enumeration, no special drivers need to be installed.
- Once the enumeration is complete, the customer can download the boot image using any standard TFTP server application.

Figure 25-25. Image Transfer for USB Boot

25.3.9.6.3 Pins Used

The list of the device pins that are configured by the ROM incase of USB boot mode are as follows. Please note that all the pins might not be driven at boot time.

Table 25-36. Pins Used for USB Boot

Signal name	Pin Used in Device
dm	usb0_dm
dp	usb0_dp
id	usb0_id
vbus	usb0_vbus

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

25.3.9.7 ASIC ID structure

The ASIC ID size is 58 Bytes for UART and 81 Bytes for others. The fields of this structure are unused. This structure is included purely for legacy purposes.

25.3.10 Image Format

25.3.10.1 Overview

All preceding sections describe how the ROM Code searches and detects a boot image from a memory or a peripheral device type. This chapter describes the format of the boot image itself.

A boot image is basically made out of two major parts:

- The software to execute (with keys and certificate in case of **HS Device**).
- A header containing the destination address and size of the image for non XIP memory devices

The mandatory section of a boot image contains the software which will be loaded into the memory and executed (if correctly authenticated in case of **HS Device**). An overview of the image formats is shown in Figure 25-26:

Figure 25-26. Image Formats on GP and HS Devices

a) GP non - XIP Memory Booting

Used for memories which require shadowing (e.g. MMC). Image must begin with a GP header which contains information on image size and destination address.

b) GP Peripheral Booting and XIP Memory Booting

When memory device is of XIP type (e.g. NOR) the GP header is not needed and the image can contain code for direct execution. . The same image format is used for peripheral booting (where the code is transferred to internal RAM).

c) HS Memory and Peripheral Booting

An HS image always begins with a TOC which points to HS image items. A TOC and items (fits in one sector) can be concatenated at top of the secure image. Items can be located in any order in the image. This format is discussed in detail in section 3.3.

-
-

25.3.10.2 Image Format for GP Device

For a **GP Device** there is no security involved and therefore neither keys nor certificates are needed in the boot image.

When the booting memory device is non-XIP (e.g. MMC) the image must contain a small header (referred as GP header) with the size of the software to load and the destination address where to store it.

The GP header is not needed when booting from an XIP memory device (e.g. NOR) or in case of peripheral booting. In this case, the peripheral or memory booting image starts directly with executable code.

Table 25-37. GP Device Image Format

Field	non XIP device	XIP device	Size[bytes]	Description
	Offset	Offset		
Size	0000h	-	4	Size of the image
Destination	0004h	-	4	Address where to store the image / code entry point
Image	0008h	0000h	x	Executable code

Copyright © 2011, Texas Instruments Incorporated

DRAFT ONLY

Note: the “Destination” address field stands for both:

- Target address for the image copy from the non-XIP storage to the target XIP location (e.g., internal RAM or SDRAM)
- Entry point for image code

In other words the user must take care to locate the code entry point to the target address for image copy.

25.3.10.3 Image Format for HS Device

A signed **HS Device** image consists of various sub-images with the associated certificates needed for authentication purposes.

The minimal HS image must at least contain:

- The Table of Content (TOC)
- The Public Keys Certificate
- The Primary Protected Application (PPA) with its Certificate
- The Initial SW with its Certificate

Additional sub-images may be embedded within the image:

- R&D Certificate
- CH items

Figure 25-27 illustrates the structure of a generic image with mandatory and optional sub-images items.

Figure 25-27. Image Structure

25.3.10.3.1 Table of Contents

The TOC is 512 bytes long and consists of a maximum of 16 TOC items (32 bytes long each), located one after the other. If the number of TOC items is less than 16, the TOC item after the last used TOC item must be filled by FFh. Each TOC item contains information required by the ROM Code such as item location, size and name, as illustrated in Figure 25-28 and Table 25-38.

Figure 25-28. TOC Structure

Table 25-38. TOC Item Fields

Offset	Field	Size[bytes]	Description
0000h	Start	4	Offset from the start address of TOC to the actual address of sub image.
0004h	Size	4	Size of sub image.
0008h	Flags	4	Reserved
000Ch	Align	4	Reserved
0010h	Load Address	4	For an Initial SW TOC item during Memory Booting this field is the target address for item copy.It is ignored in other cases.
0014h	Filename	12	12 character long name of sub image, including the zero (‘\0’) terminator.

The ROM Code recognizes sections pointed to by the TOC based on the filename as described in Table 25-39:

- The ‘X-LOADER’, ‘2ND', ‘MLO’, or ‘ULO’ section contains the Initial Software
- The ‘KEYS’ section contains the Public Keys Certificate, which must be imported in the Secure RAM for subsequent authentication.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

- The ‘R&D’ section contains the R&D Certificate, which must be authenticated before being used by the ROM Code in order to activate or deactivate debug features. Note: If Authentication fails for R&D the whole Authentication process is not stopped, only debug features are not activated.
- The ‘PRIMAPP’ section contains the PPA and its Certificate, which must be imported in the Secure RAM for subsequent authentication. If the Certificate is successfully authenticated, the PPA can be executed.
- Optionally, TOC may contain CH sections. The “CHSETTINGS” is a mandatory section of CH and is used to recognize CH presence.

Table 25-39. Filenames in TOC

Filename	Description
X-LOADER	Bootstrap
2ND	Secondary loader (Peripheral booting)
MLO	MMC/SD Loader (Memory booting)
ULO	Alternative filename used for Secondary Loader via USB (Peripheral booting)
KEYS	Contains the Public Keys Certificate
R&D	Contains the R&D Certificate
PRIMAPP	Contains the PPA and its Certificate
CHSETTINGS	Configuration Header general setting item. This item is mandatory for CH
CHRAM	Configuration Header EMIF item
CHFLASH	Configuration Header GPMC item
CHMMCSD	Configuration Header MMC/SD item

25.3.10.3.2 Certificates

For a detailed description of certificates and other remaining items of an HS image, see.

25.3.11 Authentication and Execution

25.3.11.1 Overview

One of the early steps of the Public ROM Code execution is to search for a boot image from the requested medium (configured by the SYSBOOT pins) and copy it to RAM if needed. If the device is of GP type and boot interface is non-XIP then the image is simply copied to the provided destination address (internal or external RAM) and then executed. If the boot interface is of XIP type then the image copy is not needed and execution is directly given to the XIP memory. In the case of an **HS Device** the next step of the booting procedure is to authenticate the image and to execute it with help of Secure ROM Code services.

For an **HS Device** the ROM Code detects and parses the TOC. It searches for a main TOC item which filename must be one of ‘X-LOADER’, ‘2ND’, ‘MLO’, or ‘ULO’. After detection of the main TOC item the ROM Code starts the authentication phase as shown in Figure 30 and uses the TOC to locate the different items needed for the authentication phase.

The ROM Code searches for the mandatory Public Keys certificate and imports it to secure world. If an R&D certificate is present then the requested debug features are enabled accordingly. The Primary Protected Application is authenticated and imported into Secure RAM. The purpose of the last step is to authenticate the Initial SW.

If any of the secure items fails to authenticate then the ROM Code does not start the external code execution and branches to a dead loop (with the exception of the R&D certificate).

Figure 25-29. HS Image Authentication Process

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

25.3.11.2 Execution

The image is executed at the time the ROM Code performs the branch to the first executable instruction inside the Initial SW. For a **GP Device** in non-XIP the execution address is the first word after the GP header. On an **HS Device** the execution address is a parameter of the Initial SW certificate. The branch is performed in public ARM supervisor mode. The R0 register points to the Booting Parameters structure which contains various information about the booting execution. Table 25-40 details this structure.

Table 25-40. Booting Parameters Structure

Offset	Field	Size[bytes]	Description
00h	Reserved	4	Reserved
04h	Memory booting device descriptor address	4	Pointer to the memory device descriptor that has been used during the memory booting process
05h	Current Booting Device	1	Code of device used for booting00h – void, no device 01h – XIP MUX1 memory02h – XIPWAIT MUX 1 (wait monitoring on)03h – XIP MUX2 memory04h – XIPWAIT MUX 2 (wait monitoring on)05h – NAND06h – NAND with I2C08h – MMC/SD port 009h – MMC/SD port 115h – SPI41h – UART044h – USB46h – CPGMAC0
06h	Reset Reason	1	Current reset reason bit mask (bit=1-event present)[0] – power-on reset[1] – global software reset[3] – security violation reset[4] – MPU watchdog reset[5] – secure watchdog reset[6] – external warm resetOther bits – reserved
07h	Reserved	1	Reserved

25.3.12 Wakeup

25.3.12.1 Overview

The device supports a comprehensive set of low-powered states to meet aggressive power budget requirements. The different low-power modes supported, from the lowest power consumption state to the highest power consumption state are

- RTC Only
- Deep Sleep 0
- Deep Sleep 1
- Deep Sleep 2
- Standby
- Active

Of these states, the ROM code come into the picture only in the first three, as the A8 registers (particularly the PC) are preserved in all the other states. So for these modes, on wake up, execution will resume from the next instruction following the WFI.

In the RTC only mode, the ROM code involvement is trivial, as a wakeup from the RTC only mode, for the ROM perspective, is indistinguishable from a power on reset. In Deep Sleep 0 and Deep Sleep 1 state, it is the ROM codes responsibility to detect wakeup has occurred, and branch to a user defined return address, rather than doing a full boot.

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

25.3.12.2 Wakeup Booting by ROM code

In this device, when the A8 is in OFF mode, execution begins from the reset vector on wakeup. As the reset vector lies in the ROM code, the ROM code is the first software entity that takes control of the A8 on a wakeup.

It must be noted here that in the device, in all the modes other than RTC only, the A8 internal SRAM and the L3 OCMC RAM are held in retention. This is a fundamental assumption of the ROM code. As the contents of these RAMs are not lost in the Deep Sleep modes, it is possible to return to a location in these memories.

This does away with the need of the ROM having to restore PLL and EMIF settings, which would have been needed if the ROM had to return to an DRAM address, as DRAM is held in self refresh. The job of dialing up the PLLs and restoring EMIF and other peripheral register values is the responsibility of the user code. The recommendation is that such restoration code be placed in the OCMC RAM so that the wakeup procedure is compatible between secure and non-secure devices.

In the past, having complex PLL and EMIF restore code in the ROM has made wakeup debugging very complicated. This has also traditionally been a source of many bugs, as the wakeup procedures are complicated and difficult to test exhaustively in pre-silicon.

The current architecture does away with these problems. The flow in the ROM on wakeup is kept simple and to the minimal. It only involves identifying that the reset reason is a Wakeup, and then branching to a return address, rather than proceeding with a full boot. This also helps in improving the time required to wakeup a system that is in Deep Sleep.

Figure 25-30. Wakeup Booting by ROM

25.3.13 Tracing

Tracing in the Public ROM Code consist in three 32 bit vectors for which each bit corresponds to a particular “way point” in the ROM Code execution sequence (see Table 25-4). Tracing vectors are initialized at the very beginning of the startup phase and updated all along the boot process.

There are two sets of tracing vectors. The first set is the current trace information (after cold or warm reset). The second set holds a copy of trace vectors collected at the first ROM Code run after cold reset. As a consequence after a warm reset it is possible to have visibility on the boot scenario that occurred during cold reset.

Table 25-41. Tracing Vectors

Trace vector	Bit #	Group	Meaning
1	0	General	Passed the public reset vector
1	1	General	Entered main function
1	2	General	Running after the cold reset
1	3	Boot	Main booting routine entered
1	4	Memory Boot	Memory booting started
1	5	Peripheral Boot	Peripheral booting started
1	6	Boot	Booting loop reached last device
1	7	Boot	GP header or HS TOC found
1	8	Boot	Reserved
1	9	Boot	Reserved
1	10	Peripheral Boot	Reserved
1	11	Peripheral Boot	Reserved
1	12	Peripheral Boot	Device initialized
1	13	Peripheral Boot	Asic Id sent
1	14	Peripheral Boot	Image received
1	15	Peripheral Boot	Peripheral booting failed

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 25-41. Tracing Vectors (continued)

Trace vector	Bit #	Group	Meaning
1	16	Peripheral Boot	Booting Message not received (timeout)
1	17	Peripheral Boot	Image size not received (timeout)
1	18	Peripheral Boot	Image not received (timeout)
1	19	Reserved	Reserved
1	20	Configuration Header	CHSETTINGS found
1	21	Configuration Header	CHSETTINGS executed
1	22	Configuration Header	CHRAM executed
1	23	Configuration Header	CHFLASH executed
1	24	Configuration Header	CHMMCSD clocks executed
1	25	Configuration Header	CHMMCSD bus width executed
1	26	Reserved	Reserved
1	27	Reserved	Reserved
1	28	Reserved	Reserved
1	29	Reserved	Reserved
1	30	Reserved	Reserved
1	31	Reserved	Reserved
2	0	Companion chip	Reserved
2	1	Companion chip	Reserved
2	2	Companion chip	Reserved
2	3	Companion chip	Reserved
2	4	USB	USB connect
2	5	USB	USB configured state
2	6	USB	USB VBUS valid
2	7	USB	USB session valid
2	8	Reserved	Reserved
2	9	Reserved	Reserved
2	10	Reserved	Reserved
2	11	Reserved	Reserved
2	12	Memory Boot	Memory booting trial 0
2	13	Memory Boot	Memory booting trial 1
2	14	Memory Boot	Memory booting trial 2
2	15	Memory Boot	Memory booting trial 3
2	16	Memory Boot	Execute GP image
2	17	Peripheral Boot	Start authentication of peripheral boot image
2	18	Memory & Peripheral Boot	Jumping to Initial SW
2	19	Memory & Peripheral Boot	High Secure image not executed
2	20	Memory & Peripheral Boot	Start image authentication
2	21	Memory & Peripheral Boot	Image authentication failed
2	22	Memory & Peripheral Boot	Analyzing SpeedUp
2	23	Memory & Peripheral Boot	Speedup failed
2	24	Memory & Peripheral Boot	PPA authentication success
2	25	Memory & Peripheral Boot	PPA authentication failed
2	26	Memory & Peripheral Boot	R&D authentication success
2	27	Memory & Peripheral Boot	R&D authentication failed

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

Table 25-41. Tracing Vectors (continued)

Trace vector	Bit #	Group	Meaning
2	28	Memory & Peripheral Boot	Authentication procedure failed
2	29	Reserved	Reserved
2	30	Reserved	Reserved
2	31	Reserved	Reserved
3	0	Memory Boot	Memory booting device NULL
3	1	Memory Boot	Memory booting device XIP
3	2	Memory Boot	Memory booting device XIPWAIT
3	3	Memory Boot	Memory booting device NAND
3	4	Memory Boot	Memory booting device OneNAND
3	5	Memory Boot	Memory booting device MMCSD0
3	6	Reserved	Reserved
3	7	Memory Boot	Memory booting device MMCSD1
3	8	Reserved	Reserved
3	9	Reserved	Reserved
3	10	Memory Boot	Memory booting device EMIF LPDDR2-NVM
3	11	Reserved	Memory booting device SPI
3	12	Reserved	Reserved
3	13	Reserved	Reserved
3	14	Reserved	Reserved
3	15	Reserved	Reserved
3	16	Reserved	Peripheral booting device UART0
3	17	Reserved	Peripheral booting device UART1
3	18	Peripheral Boot	Peripheral booting device UART2
3	19	Reserved	Reserved
3	20	Peripheral Boot	Peripheral booting device USB
3	21	Peripheral Boot	Peripheral booting device USB ULPI
3	22	Peripheral Boot	Peripheral booting device GPGMAC0
3	23	Reserved	Peripheral booting device PCIe
3	24	Reserved	Peripheral booting device NULL
3	25	Reserved	Reserved
3	26	Reserved	Reserved
3	27	Reserved	Reserved
3	28	Reserved	Reserved
3	29	Reserved	Reserved
3	30	Reserved	Reserved
3	31	Reserved	Reserved

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated

25.4 Use Cases

25.5 Registers

DRAFT ONLY

Copyright © 2011, Texas Instruments Incorporated